TUGAS AKHIR – SS 141501

ANALISIS TWITTER PELANGGAN BELANJA *ONLINE* MENGGUNAKAN METODE *NAÏVE BAYES CLASSIFIER* (NBC) DAN *ARTIFICIAL NEURAL NETWORK* (ANN)

FRANSISKA KRISTIN DAMAYANTI
NRP 062116 4500 0035

Dosen Pembimbing
Dr. Kartika Fithriasari, M.Si

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018

TUGAS AKHIR – SS141501

ANALISIS TWITTER PELANGGAN BELANJA *ONLINE* MENGGUNAKAN METODE *NAÏVE BAYES CLASSIFIER* (NBC) DAN *ARTIFICIAL NEURAL NETWORK* (ANN)

FRANSISKA KRISTIN DAMAYANTI
NRP 062116 4500 0035

Dosen Pembimbing
Dr. Dra. Kartika Fithriasari, M.Si

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018

FINAL PROJECT – SS141501

TWITTER ANALYSIS FOR *ONLINE* SHOPPING CONSUMERS USING NAÏVE BAYES CLASSIFIER (NBC) AND ARTIFICIAL NEURAL NETWORK (ANN)

FRANSISKA KRISTIN DAMAYANTI
SN 062116 4500 0035

Supervisors
Dr. Dra. Kartika Fithriasari, M.Si

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018

LEMBAR PENGESAHAN

ANALISIS TWITTER UNTUK KONSUMEN BELANJA *ONLINE* MENGGUNAKAN METODE *NAÏVE BAYES CLASSIFIER* (NBC) DAN *ARTIFICIAL NEURAL NETWORK* (ANN)

TUGAS AKHIR
Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
pada
Program Studi Sarjana Departemen Statistika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

Oleh :
Fransiska Kristin Damayanti
NRP. 062116 4500 0035

Disetujui oleh Pembimbing:

Dr. Dra. Kartika Fithriasari, M.Si ( )
NIP. 19691212 199303 2 002

Mengetahui,
Kepala Departemen

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2018

ANALISIS TWITTER UNTUK KONSUMEN BELANJA *ONLINE* MENGGUNAKAN METODE *NAÏVE BAYES CLASSIFIER* (NBC) DAN *ARTIFICIAL NEURAL NETWORK* (ANN)

Nama Mahasiswa : Fransiska Kristin Damayanti
NRP : 062116 4500 0035
Departemen : Statistika
Dosen Pembimbing : Dr. Dra. Kartika Fithriasari, M.Si

Abstrak

Tokopedia dan Bukalapak merupakan perusahaan yang mengusung bisnis marketplace dan mall online. Untuk menanggapi pendapat, kritik, saran dan masalah complain, Tokopedia dan Bukalapak mempunyai akun khusus pada Twitter yang diberi nama @tokopediacare dan @bukabantuan. Jenis komentar tersebut berbentuk unstructured text dalam jumlah besar. Kondisi ini dapat mengakibatkan perusahaan belanja online dapat melewatkan informasi yang berguna dari sekumpulan dokumen teks. Oleh karena itu, dilakukan analisis text mining dengan membandingkan metode klasifikasi Naïve Bayes Classifier (NBC) dan Artificial Neural Network (ANN). Berdasarkan hasil analisis dapat diketahui metode klasifikasi Artificial Neural Network lebih baik digunakan jika dibandingkan dengan Naïve Bayes Classifier dengan hasil ketepatan klasifikasi mencapai 0,9492 pada akun TokopediaCare dan 0,97564 akun BukaBantuan . Selain itu dilakukan pula Social Network Analysis yang digunakan untuk menggambarkan struktur komunikasi dan tingkat partisipasi dari setiap pelanggan.

Kata Kunci : Artificial Neural Network, Naïve Bayes Classifier, Sentiment Analysis, Social Network Analysis, Text Mining.

(Halaman ini Sengaja Dikosongkan)

TWITTER ANALYSIS FOR *ONLINE* SHOPPING CONSUMERS USING NAÏVE BAYES CLASSIFIER (NBC) AND ARTIFICIAL NEURAL NETWORK (ANN)

Student Name : Fransiska Kristin Damayanti
Student Number : 062116 4500 0035
Department : Statistics
Supervisors : Dr. Dra. Kartika Fithriasari, M.Si

Abstract

Tokopedia and Bukalapak is a company engaged in marketplace and online mall. To respond to opinions, critics and complaints, Tokopedia and Bukalapak have a special account on Twitter called @tokopediacare and @bukabantuan. The comment type is unstructured text in large numbers. This condition can lead to online shopping companies skipping useful information from a collection of text documents. Therefore, text mining analysis is used by comparing Naïve Bayes Classifier (NBC) and Artificial Neural Network (ANN) classification method. Based on the analysis results, it is known that ANN classification method.is better used when compared with NBC classification method, with classification accuracy reach 0.9492 for TokopediaCare and 0.9764 for BukaBantuan. In addition, Social Network Analysis (SNA) is also used to describe the structure of communication and the level of participation of each customer.

Keyword : Artificial Neural Network, Naïve Bayes Classifier, Sentiment Analysis, Social Network Analysis, Text Mining.

(Halaman ini Sengaja Dikosongkan)

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan kenikmatan, kemudahan serta karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan judul **"Analisis Twitter Pelanggan Belanja *Online* Menggunakan Metode *Naïve Bayes Classifier* (NBC) dan *Artificial Neural Network* (ANN)"**

Terselesaikannya Tugas Akhir ini tak lepas dari peran serta berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih dengan penuh hormat dan kerendahan hati, kepada:

1. Ibu Dr. Dra. Kartika Fithriasari, M.Si selaku dosen pembimbing yang selalu memberikan dukungan, pelajaran, masukan dan telah sabar membimbing penulis dalam penyusunan Tugas Akhir.
2. Bapak Prof. Drs. Nur. Iriawan, M.IKom, Ph.D dan Ibu Pratnya Paramitha Oktaviana, S.Si, M.Si selaku dosen penguji yang telah memberikan kritik dan saran yang membangun untuk menyempurnakan Tugas Akhir.
3. Bapak Dr. Suhartono selaku Kepala Departemen Statistika FMKSD ITS yang telah menyediakan fasilitas untuk menyelesaikan Tugas Akhir.
4. Bapak Dr. Sutikno, M.Si selaku Koordinator Program Studi S1 atas ketelatenannya dalam memberikan bantuan dan informasi yang diberikan selama ini.
5. Jurusan Statistika ITS beserta seluruh dosen Statistika ITS yang telah memberikan ilmu-ilmu yang bermanfaat serta segenap karyawan Jurusan Statistika ITS yang melayani mahasiswa dengan sabar.
6. Keluarga besar penulis khusunya Ibu Sri Harmijati dan Bapak Sarni yang senantiasa memberikan doa, motivasi, dukungan, kepercayaan, kasih sayang dan kesabaran tiada batas dalam memberikan pelajaran hidup yang diberikan kepada penulis. Kakak penulis Anastasia Yuni W,

Andreas Didit, Robertus Hendik serta kedua keponakan Rizky dan Ica yang selalu memberi kebahagiaan.

7. Sahabat-sahabat tercinta Zuyyin Inesa P, Raras A, Rima K, Inung Anggun S, Camelia Nanda S, Siti Azizah NS, Putri Ayu Sekar K, Rakhmah Wahyu M, Yongky C, Novi Ajeng S, Nym Cista SD, Dimas Ewin A, dan Lely Presti yang selalu memberi dukungan, berbagi cerita baik suka maupun duka selama di perkuliahan dan bisa menjadi tempat curahan hati ketika penulis merasa "*low motivation*" dalam menyelesaikan Tugas Akhir.
8. Teman-teman Paduan Suara Mahasiswa ITS khususnya LA 13 dan LA 14 yang selalu memberikan dukungan, semangat, dan kebahagiaan selalu.
9. Teman-teman Lintas Jalur Statistika ITS 2016 yang telah melalui proses pembelajaran bersama-sama mulai awal perkuliahan sampai pembuatan Tugas Akhir.
10. Teman-teman IPA SMA Negeri 5 Madiun khususnya Agung, Nerindra, Ruso, Rully, Dhela, Hafid, Ma'ruf yang selalu memberikan hiburan serta dukungan saat sedang mengalami kesusahan.
11. Ignatius Chandra Setyanto yang selalu memberikan semangat dan motivasi untuk segera menyelesaikan Tugas Akhir.
12. Pihak-pihak lain yang telah mendukung dan membantu penyusunan Tugas Akhir ini yang tidak mungkin penulis sebutkan satu persatu.

Dengan selesainya laporan Tugas Akhir ini, penulis menyadari bahwa penelitian Tugas Akhir ini masih belum sempurna, jika masih ada kekurangan diharapkan saran dan kritik agar dapat mengembangkan penelitian ini.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

(Halaman ini Sengaja Dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Informasi merupakan hal yang sangat penting dalam kehidupan bermasyarakat. Beragam informasi dapat membuat pengetahuan seseorang menjadi luas. Perkembangan ilmu dan teknologi informasi telah banyak mengubah cara pandang dan gaya hidup masyarakat Indonesia dalam menjalankan kegiatannya. Di negara berkembang, *mobile phone* membantu pasar perdagangan untuk mengejar ketertinggalan dengan cepat. The Tetra Pak Index 2017 dan Laporan Tahunan 2016 Kementerian Komunikasi dan Informatika menunjukkan ada sekitar 132 juta pengguna internet di Indonesia yang terdiri dari 52,5% laki-laki dan 47,5% perempuan. Hal ini menunjukkan bahwa terdapat lebih dari 50% penduduk Indonesia telah terhubung dengan internet. Perkembangan teknologi informasi yang sangat cepat memberikan perkembangan pula pada dunia bisnis dan pemasaran salah satunya adalah bisnis *online*.

Internet perlahan-lahan mulai menggeser budaya pembelian dari cara konvensional menjadi lebih modern atau disebut *online shopping*. *Online shopping* adalah pembelian yang dilakukan via Internet sebagai media pemasaran dengan menggunakan website sebagai katalog. Salah satu kelebihan *online shopping* yaitu selain pembeli bisa melihat desain produk yang sudah ada, pembeli juga bisa menentukan desain hingga melakukan pembayaran secara *online*. Bisnis ini memanfaatkan para pengguna Internet sebagai target konsumen mereka (Ollie, 2008).

PT. Tokopedia merupakan salah satu *mall online* di Indonesia yang mengusung model bisnis *marketplace* dan *mall online*. Wujud sebuah *mall online* yang mempertemukan penjual dan pembeli dan memungkinkan untuk terjadinya transaksi jual beli *online* dengan aman dan nyaman. Bergabung untuk menggunakan Tokopedia sangatlah mudah dan tidak dipungut

biaya. Setelah beroperasi www.tokopedia.com telah menjadi salah satu *online marketplace* dengan tingkat pertumbuhan yang sangat pesat di Indonesia walaupun usianya masih seumur jagung, baik dalam jumlah anggota, toko, *online* aktif, jumlah produk hingga jumlah transaksi pembelian dan penjualan setiap harinya. Tokopedia sejatinya tidak mempunyai cabang perusahaan. Selain Tokopedia, salah satu e-commerce yang saat ini sedang tinggi penjualannya adalah Bukalapak. Bukalapak merupakan salah satu situs jual-beli *online* yang didirikan pada tahun 2015 oleh Ahmad Zaky dan hingga saat ini menjadi sarana bagi para pemilik usaha dapat membuka toko *online* dan melayani pembeli dari seluruh Indonesia untuk jumlah transaksi satuan maupun grosir. Produk yang dapat diperdagangkan di Bukalapak berupa barang yang aman dan terjamin kualitasnya, serta bisa dikirim melalui jasa pengiriman. Kedua marketplace ini memungkinkan setiap individu dan pemilik bisnis di Indonesia untuk mengembangkan dan mengelola bisnis *online* mereka secara mudah dan gratis, sekaligus memungkinkan pengalaman berbelanja *online* yang lebih aman dan nyaman.

Belanja *online* pada saat ini memanfaatkan situs-situs jejaring sosial seperti Youtube, MySpace, Facebook, Instagram dan Twitter. Keunggulan media sosial dibandingkan media konvensional adalah *feedback* secara terbuka, saling berkomentar dalam waktu cepat dan tidak terbatas. Melihat kondisi tersebut, banyak perusahaan dan perorangan yang kemudian memanfaatkannya sebagai media untuk melakukan promosi yang dirasa efektif dan efisien dibanding media konvensional. Sebagian besar layanan situs sosial tersebut berdasarkan web (*web based*) dan menyediakan fasilitas bagi pengguna untuk berinteraksi dengan pengguna lain. Twitter adalah aplikasi sosial media yang masih menyediakan fasilitas '*search*' ke seluruh status/twit yang dimilikinya. Facebook dan Instagram hanya menyediakan akses terhadap Public Page, sedangkan WhatsApp tidak dapat ditangkap percakapan di dalamnya.

Pengguna Twitter sendiri bisa terdiri dari berbagai macam kalangan yang para penggunanya ini dapat berinteraksi dengan teman, keluarga hingga rekan kerja. Twitter sebagai sebuah situs jejaring sosial memberikan akses kepada penggunanya untuk mengirimkan sebuah pesan singkat yang terdiri dari maksimal 140 karakter (disebut tweet). Tweet sendiri bisa terdiri dari pesan teks dan foto. Melalui tweet inilah pengguna Twitter dapat berinteraksi lebih dekat dengan pengguna Twitter lainnya dengan mengirimkan tentang apa yang sedang mereka pikirkan, apa yang sedang dilakukan, tentang kejadian yang baru saja terjadi, tentang berita terkini serta hal lainnya. Twitter sendiri telah menyediakan fasilitas Twitter API yang memberikan kemudahan untuk para peneliti untuk mengkoleksi dan mengumpulkan tweet. Twitter API memfasilitasi pengguna untuk dapat mengirimkan *request query* sebanyak 180 request/15 menit. Jika sebelum waktu 15 menit, *request* telah mencapai 180, maka harus menunggu 15 menit berikutnya untuk bisa melakukan *request* kembali (Kumar, Morstatter, & Liu, 2013).

Tokopedia dan Bukalapak mempunyai sebuah akun Twitter *costumer care* yaitu @tokopediacare untuk Tokopedia dan @bukabantuan untuk Bukalapak. Akun tersebut adalah salah satu bentuk layanan pelanggan khusus melalui *online* yang disediakan untuk menanggapi tanggapan, pendapat, kritik, saran dan masalah *complain.* Komentar dalam twitter berbentuk teks, sehingga perlu dilakukan analisis *text mining*. *Text mining* dapat memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian/ pengelompokan dan menganalisa *unstructured text* dalam jumlah besar. Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen dan mendukung proses *knowledge discovery* pada koleksi dokumen yang besar, jadi sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak teratur atau minimal semi teratur (Gupta dan Lehal, 2009). Sebelum suatu dokumen dianalisis menggunakan metode-metode dalam *text mining*, harus dilakukan tahapan *pre-processing* teks dahulu. *Pre-processing* teks yang

dilakukan yaitu meliputi proses *tokenizing*, *case folding*, *stemming*, dan *filtering stopwords*. Hasil *pre-processing* teks adalah berupa data dengan dokumen atau artikel dianggap sebagai observasi dan kata kunci yang didapatkan dianggap sebagai atribut atau variabel. Dari proses ini, tentunya akan terbentuk suatu data dengan jumlah atribut sangat banyak karena kata kunci yang terkandung dalam satu dokumen bisa lebih dari satu. Oleh karena itu, hasil *pre-processing* teks tersebut dapat dianggap sebagai *big data*. Data yang dihasilkan dari *pre-processing* teks ini diharapkan telah siap diolah dengan metode *text mining*.

Mengetahui tanggapan dari masyarakat khususnya penjual dan pembeli terhadap layanan Tokopedia dan Bukalapak merupakan hal yang tidak dapat dikesampingkan. Informasi yang didapatkan dari twitter tidak dapat menggambarkan struktur komunikasi dan tingkat partisipasi dari setiap pelanggan. *Social Network Analysis* (SNA) adalah salah satu metode yang digunakan dalam penelitian ini yang bermanfaat untuk mengetahui pola interaksi, struktur komunikasi dan tingkat partisipasi dari setiap pelanggan (Alexander dkk, 2011). Salah satu penerapan *text mining* yang banyak dilakukan adalah untuk otomatisasi klasifikasi dokumen. Dalam proses klasifikasi, terdapat dua jenis data, yaitu data yang digunakan untuk pelatihan model (*training*) dan data yang digunakan untuk pengujian model (*testing*). Proses pelatihan model diharapkan dapat memberikan sebuah model yang optimum, sehingga baik digunakan untuk mengategorikan data dalam beberapa kelompok. Setelah didapatkan model, proses selanjutnya adalah dilakukan pengujian. Proses pengujian dilakukan terhadap data uji (*testing*), yaitu dengan mengaplikasikan model yang telah didapatkan dari hasil perhitungan menggunakan data *training*.

Penelitian terdahulu tentang *marketplace* yang telah dilakukan Dita Novita (2016) menghasilkan kesimpulan bahwa dari hasil perseptual map ditunjukkan titik koordinat Tokopedia dan Bukalapak sangat berdekatan yang berarti terjadi persaingan yang sangat ketat antara kedua *marketplace* tersebut. Pengelompokan atau klasifikasi teks saat ini telah dilakukan,

diantaranya adalah penelitian yang dilakukan oleh Dwi Ary dan Kartika Fithriasari (2016) yang membandingkan metode SVM, ANN, dan NBC. Dari penelitian tersebut dihasilkan bahwa metode NBC memberikan hasil paling baik. Menurut Vapnik (1999), ANN memiliki kelebihan dalam hal kemampuan untuk generalisasi, yang bergantung pada seberapa baik ANN meminimalkan resiko empiris, yaitu faktor kesalahan pada saat *training*. Sebagai komparasi, dalam penelitian ini digunakan metode NBC karena metode ini merupakan salah satu metode yang paling banyak digunakan untuk klasifikasi data, khususnya data teks. Menurut Darujati dan Gumelar (2012), algoritma NBC memiliki kelebihan dalam hal keefektifan kategorisasi teks, algoritma yang digunakan cukup sederhana, cepat, dan memiliki akurasi yang tinggi.

1.2 Rumusan Masalah

Konsumen dari belanja *online* dapat memberikan *feedback* secara terbuka, saling berkomentar dalam waktu cepat dan tidak terbatas terhadap pelayanan dari Tokopedia dan Bukalapak melalui twitter *costumer care*. Jenis komentar tersebut berbentuk *unstructured text* dalam jumlah besar. Kondisi ini dapat mengakibatkan perusahaan belanja *online* melewatkan informasi yang berguna dari sekumpulan dokumen teks. Mengetahu*i sentiment* dari konsumen belanja *online* secara manual dapat merugikan waktu, dan tenaga. Oleh karena itu, pada penelitian ini dilakukan analisis *text mining* dengan membandingkan metode klasifikasi *Naïve Bayes Classifier* (NBC) dan *Artificial Neural Network* (ANN).

Informasi yang didapatkan dari twitter tidak dapat menggambarkan struktur komunikasi dan tingkat partisipasi dari setiap pelanggan. Oleh karena itu diperlukan suatu metode yang dapat menilai atau memeriksa pola interaksi pelanggan Tokopedia dan Bukalapak. *Social Network Analysis* (SNA) merupakan salah satu metode untuk menganalisis pola interaksi pelanggan Tokopedia dan Bukalapak.

1.3 Tujuan

Berdasarkan rumusan pada penelitian ini, tujuan yang akan dicapai dalam penelitian ini adalah mengetahui metode klasifikasi terbaik antara Naïve Bayes Classifier (NBC) dan Artificial Neural Network (ANN). Selain itu, juga ingin mengetahui struktur komunikasi dan tingkat partisipasi dari pelanggan tokopedia dan Bukalapak serta mendapatkan kata-kata yang sering muncul pada masing-masing sentimen menggunakan visualisasi *wordcloud*.

1.4 Manfaat

Manfaat yang dapat diperoleh dari penelitian ini adalah mampu mengetahui proses mengekstrak dan mengolah data teks secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini dengan hasil metode klasifikasi terbaik. Selain itu juga dapat mengidentifikasi pola pengguna twitter konsumen bukalapak dan tokopedia sehingga dapat dimanfaatkan lebih maksimal untuk menyebarkan informasi secara lebih efektif.

1.5 Batasan Masalah

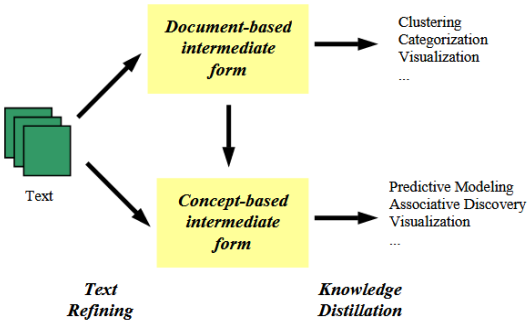
Pada penelitian ini, batasan masalah yang digunakan adalah sebagai berikut.

1. Tweet yang digunakan untuk analisis diambil dari @tokopediacare dan @bukabantuan.
2. Penelitian ini hanya melakukan analisis sentimen terhadap *tweet* berbahasan Indonesia.
3. Penelitian ini tidak mengatasi ata atau kalimat yang cara penulisannya tidak umum (disingkat)
4. Data twitter yang digunakan merupakan tweet yang diunggah pada tanggal 6 Februari 2018 hingga 6 Maret 2018.
5. Sentimen awal yang digunakan adalah sentimen positif dan negatif yang ditentukan secara subjektif oleh peneliti.

BAB II
TINJAUAN PUSTAKA

2.1 *Text Mining*

Menurut Ronen Feldman dan James Sanger dalam buku *The Text mining Handbook*, *text mining* dapat didefinisikan secara luas sebagai suatu proses menggali informasi dimana seorang user berinteraksi dengan sekumpulan dokumen menggunakan *tools analysis* yang merupakan komponen-komponen *data mining* yang salah satunya adalah kategorisasi. *Text mining* memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian/ pengelompokan dan menganalisa *unstructured text* dalam jumlah besar. Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen dan mendukung proses *knowledge discovery* pada koleksi dokumen yang besar, jadi sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak teratur atau minimal semi teratur. Adapun tugas khusus dari *text mining* antara lain yaitu pengkategorisasian teks (*text categorization*) dan pengelompokan teks (*text clustering*) (Fieldman & Sanger, 2006).



Gambar 2.1 Kinerja *Text mining* (Sumber: Tan, 1999)

Pada Gambar 2.1 menunjukkan kerangka kerja dari *text mining*. Pada awal ditempuh langkah *text refining* yaitu

pengubahan bentuk dari teks asli menjadi bentuk *intermediate* (*intermediate form*), yang dapat berbasis dalam bentuk dokumen (*document-based intermediate form*) atau berbasis pada konsep (*concept-based intermediate form*). Tahap berikutnya adalah tahap *knowledge distillation*. Pada tahap ini jika bentuk intermediate berupa dokumen maka kegiatan distilasi pengetahuan dapat berupa kegiatan mengelompokkan dokumen, kategorisasi dokumen, visualisasi dan lain sebagainya. Pada bentuk intermediate berupa konsep kegiatan distilasi dapat berupa *predictive modeling*, *assosiative discovery* dan visualisasi. Salah satu kegiatan penting dalam distilasi pengetahuan adalah klasifikasi atau kategorisasi teks dengan pendekatan *supervised learning*. Kategorisasi teks sendiri saat ini memiliki berbagai cara pendekatan antara lain berbasis numeris, misalnya pendekatan *probabilistic, support vector machine,*dan *artificial neural network*, serta berbasis non numeris seperti *decision tree classification* (Tan, 1999)

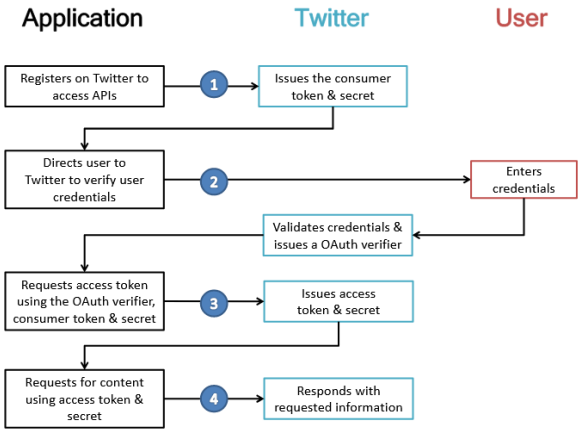
2.2 API (Application Programming Interface)

Sebelum melakukan pengolahan data, langkah pertama yang harus dilakukan adalah *crawling* Twitter data dengan menggunakan API (*Application Programming Interface*). Kegunaan dari API adalah untuk mengetahui informasi tentang pengguna, jaringan pengguna yang terdiri dari koneksi, dan tweet yang di bicarakan (Kumar, Morstatter, & Liu, 2013).

API adalah sekumpulan perintah, fungsi, dan protocol yang dapat digunakan saat membangun perangkat lunak untuk system operasi tertentu. API memungkinkan *programmer* untuk menggunakan fungsi standar untuk berinteraksi dengan system informasi. API juga merupakan suatu dokumentasi yang terdiri dari antar muka, fungsi, kelas, struktur untuk membangun sebuah perangkat lunak sehingga dapat memudahkan seorang programmer untuk membongkar suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya. Suatu rutin standar yang memungkinkan

developer menggunakan *system function*. Proses ini dikelola melalui *operating system*. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya untuk saling berinteraksi.

Twitter API hanya dapat diakses melalui permintaan otentik. Twitter menggunakan *Open Authentication* dan setiap permintaan harus ditandatangani oleh pengguna Twitter yang valid. Akses ke Twitter API juga terbatas pada sejumlah permintaan dan batas waktu tertentu. Pembatasan jumlah Twitter ini diterapkan secara baik pada pengguna individual maupun pada tingkat aplikasi (Kumar, Morstatter, & Liu, 2013). Berikut adalah gambar langkah-langkah *crawling* data Twitter melalui Twitter API.



Gambar 2.2 *OAuth Workflow* (Sumber: Kumar, Morstatter, & Liu, 2013).

2.3 Praproses Data

Menurut Septiana, Ridok, dan Dewi (2013), *pre-processing* teks merupakan tahapan awal dalam pengolahan teks yang digunakan untuk pengubahan bentuk dokumen menjadi data terstruktur sesuai kebutuhannya dan mendapatkan nilai numerik dari kata untuk dapat diolah lebih lanjut dalam proses *text mining*. Menurut Indriani (2014), *pre-processing* dalam proses klasifikasi dokumen digunakan untuk membangun sebuah *index* dari koleksi

dokumen. Tujuan *pre-processing* adalah untuk meningkatkan akurasi data (Rifqi, Maharani, dan Shaufiah, 2011). *Pre-processing* dalam *text mining* cukup kompleks bila dibandingkan dengan *pre-processing* untuk data angka atau kategori. Hal ini dikarenakan struktur kalimat atau teks sangatlah rumit.

Aturan penulisan dalam Bahasa Indonesia cukup ketat. Hal ini ditandai salah satunya adalah dengan adanya aturan penulisan kalimat baku yang harus sesuai dengan tata bahasa yang ada. Setiap jenis kalimat yang ada dalam Bahasa Indonesia, memiliki struktur penulisan yang berbeda-beda. Terdapat empat macam kata imbuhan (afiks) yang dapat digunakan untuk mengubah makna kata dasar dalam Bahasa Indonesia, antara lain adalah sebagai berikut.

a. Awalan (prefiks), yaitu imbuhan yang ditambahkan di depan kata dasar. Imbuhan jenis ini dibagi menjadi dua jenis, yaitu standar (‘di-‘, ‘ke-‘, dan ‘se-‘) dan kompleks (‘me-‘, ‘be-‘, ‘pe-‘, dan ‘te-‘).
b. Akhiran (sufiks), yaitu imbuhan yang ditambahkan di belakang kata dasar. Sufiks yang biasanya digunakan adalah ‘–i’, ‘-kan’, ‘-an’. Sufiks dapat ditambahkan secara langsung tanpa mengubah bentuk kata dasar. Kata ganti kepemilikan (‘-ku’, ‘-mu’, dan ‘-nya’) dan partikel (‘-lah’, ‘-kah’, ‘-tah’, dan ‘-pun’) juga dapat dikategorikan sebagai sufiks.
c. Awalan dan akhiran (konfiks), yaitu imbuhan yang ditambahkan di depan dan belakang kata dasar. Imbuhan ini dapat juga dianggap sebagai prefiks dan sufiks yang ditambahkan pada kata dasar secara bersama-sama
d. Sisipan (infiks), yaitu imbuhan yang ditambahkan di tengah-tengah kata dasar.

Konsep kata dasar dalam Bahasa Indonesia sangat berkaitan dengan proses *pre-processing* teks, karena hasil akhir proses ini diharapkan mendapatkan kata dasar yang sesuai dengan kata dalam Kamus Besar Bahasa Indonesia. Untuk mendapatkan *output* kata seperti yang diharapkan, dilakukan beberapa langkah dalam *pre-processing* teks. Secara sederhana, terdapat dua langkah penting

dalam *pre-processing* teks, yaitu proses normalisasi dan pembersihan kata (Carvalho, Matos, dan Rocio, 2007). Secara lebih terperinci, tahapan dalam *pre-processing* data teks adalah sebagai berikut.

1. *Case Folding*, merupakan proses untuk mengubah semua karakter teks menjadi huruf kecil serta menghilangkan tanda baca dan angka. Cara kerja *case folding* adalah memproses huruf alphabet dari "a" hingga "z" saja sehingga karakter selain huruf tersebut akan dihilangkan seperti tanda baca titik (.), koma (,), dan angka (Weiss, 2010).
2. *Cleansing*, merupakan proses membersihkan *tweet* dari kata yang tidak diperlukan untuk mengurangi *noise*. Kata yang dihilangkan dalam Twitter adalah karakter HTML, *emoticons*, *hastag (#)*, *username (@username)*, dan *URL*.
3. *Tokenizing*, merupakan proses memecah yang semula berupa kalimat menjadi kata-kata atau memutus urutan string menjadi potongan-potongan seperti kata-kata berdasarkan tiap kata yang menyusunnya. Sehingga dapat dikatakan mengembalikan kata pengubung.
4. *Stopwords*, merupakan kosakata yang bukan termasuk kata unik atau ciri pada suatu dokumen atau tidak menyampaikan pesan apapun secara signifikan pada teks atau kalimat (Dragut dkk, 2009). Kosakata yang dimaksud yaitu seperti kata penghubung dan kata keterangan yang bukan merupakan kata unik, misalnya "dari", "akan", "seorang", dan sebagainya.
5. *Stemming*, merupakan proses untuk mendapatkan kata dasar dengan cara menghilangkan awalan, akhiran, sisipan, dan *confixes* (kombinasi dari awalan dan akhiran). Pada penelitian ini algoritma *stemming* yang digunakan adalah *Confix Striping Stemmer* yang merupakan pengembangan dari algoritma *Nazief and Adriani's Stemmer*

2.4 K-*fold Cross Validation*

Pada *K-fold Cross Validation*, data awal secara acak dibagi menjadi *k* himpunan bagian "*folds*" yaitu $D_1, D_2, \cdots, D_k$, masing-masing berukuran sama dengan pembagian data *training* dan data *testing* dilakukan sebanyak *k*. Pada iterasi yang pertama, partisi pada D_i digunakan sebagai data *testing*, dan pastisi yang tersisa digunakan sebagai data *training*. Sehingga, pada iterasi pertama, himpunan bagian $D_2, \cdots, D_k$ secara kolektif digunakan sebagai data *training* untuk mencari model yang pertama dan diujikan pada data *testing* (D_1). Pada iterasi kedua, data *training* yang digunakan adalah himpunan bagian $D_1, D_3, \cdots, D_k$ sedangkan D_2 sebagai data yang *testing* dan seterusnya. Untuk klasifikasi, taksiran nilai akurasi adalah jumlah keseluruhan klasifikasi yang benar dari iterasi *k*, dibagi dengan banyaknya *tuples* pada data awal (Han & Kamber, 2006)

2.5 *Feature Selection*

Feature Selection dalam *machine learning* disebut juga *variable selection* (seleksi variabel) adalah proses pemilihan variabel yang relevan untuk digunakan dalam pembentukan model. Ada berbagai keuntungan dari *feature selection* antara lain dapat menyederhanaan model sehingga lebih mudah di interpretasi oleh peneliti/pembaca dan waktu pelatihan yang lebih singkat (Yang & Pederson, 1997). Pada penelitian ini, *feature selection* yang digunakan adalah χ^2 dengan hipotesis sebagai berikut.

$H_0 : \pi_{ij} = \pi_{i+}\pi_{+j}$ (Tidak ada hubungan antara kedua variabel)

$H_1 : \pi_{ij} \neq \pi_{i+}\pi_{+j}$ (Ada hubungan antara kedua variabel)

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{\left(n_{ij} - \hat{\mu}_{ij}\right)^2}{\hat{\mu}_{ij}} \qquad (2.1)$$

Dimana :

n_{ij} = Frekuensi pada sel baris ke-*i* dan kolom ke-*j*

$\hat{\mu}_{ij}$ = Frekuensi harapan pada sel baris ke-*i* dan kolom ke-*j*

π_{i+} = Total pada baris ke-*i* atau marginal frekuensi baris ke-i

π_{+j} = Total pada kolom ke-j atau marginal frekuensi baris ke-i

r = jumlah baris

c = jumlah kolom

Jumlah derajat bebas yang digunakan dalam χ^2 *test* adalah sebagai berikut.

$$\begin{aligned} df &= (IJ-1)-(I-1)-(J-1) \\ &= (I-1)(J-1) \end{aligned} \tag{2.2}$$

Jika $\chi^2 > \chi^2_\alpha$ dengan $df = (I-1)(J-1)$ maka H_0 akan ditolak pada tingkat signifikansi yang digunakan, sebaliknya gagal menolak H_0 (Agresti, 2002).

2.6 SMOTE (*Synthetic Minority Oversampling Technique*)

Synthetic Minority Oversampling Technique (SMOTE) adalah salah satu metode *oversampling*, yang bekerja dengan menerapkan metode sampling untuk meningkatkan jumlah kelas minoritas melalui replikasi data secara acak, sehingga jumlah data kelas minor sama dengan jumlah data kelas mayor. Algoritma SMOTE pertama kali diperkenalkan oleh Nithes V. Chawla (2002). Pendekatan ini bekerja dengan membangun data tiruan pada data minor. Sampel data sintentis dibuat dengan menghitung selisih jarak antara vektor atribut yang dipilih dengan vektor tetangga yang terletak berdekatan. Setelah itu, nilai selisih tersebut dikalikan dengan angka acak antara 0 sampai 1, dan kemudian ditambahkan pada nilai atribut vektor yang sebelumnya. Pada umumnya dapat dirumuskan sebagai berikut (Sain & Purnami, 2015)

$$x_{syn} = x_i + (x_j - x_i) \times \delta \quad (2.3)$$

Keterangan :

x_{syn} = data *synthetic*

x_i = contoh dari data ke-*i* pada kelas minor

x_j = contoh dari data ke-*j* berdasarkan *k* tetangga terdekat dari data x_i

δ = angka acak diantara 0 dan 1

Algoritma SMOTE menurut Nithes V. Chawla pada tahun 2002 ditulis dalam bentuk *pseudocode* sebagai berikut.

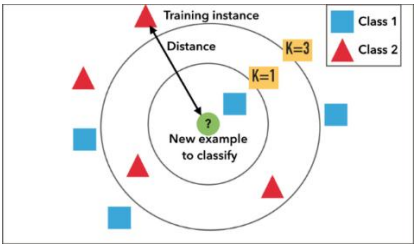
Algoritma *SMOTE* (T, N, k)
Input : T (jumlah sampel minoritas); N (persentase SMOTE); *k*-tetangga terdekat
Output : (*N*/100) * T sampel sintetis kelas minoritas

1. (* *Apabila N kurang dari 100%, randomisasi sampel kelas minoritas akan di SMOTE*)
2. **if** *N* < 100
3. **then** Randomisasi *T* sampel minoritas
4. T = (*N*/100) * T
5. N = 100
6. **endif**
7. N = (*int*)(*N*/100)(* *jumlah SMOTE diasumsikan integer dari perhitungan 100*)
8. *k* = jumlah dari tetangga terdekat
9. *numattrs* = jumlah atribut
10. *Sampel*[][] = array dari sampel kelas minoritas awal
11. *newindex* = jumlah dari sampel sintetis yang dibangkitkan, diawali dengan 0
12. *Synthetic[]*[] = array dari sampel sintetis
    (* *menghitung k tetangga terdekat untuk setiap sampel kelas minoritas*)
13. **for** *i* ← to *T*
14. Hitung k tetangga yang berdekatan pada *i* dan simpan indexnya ke *nnarray*
15. Populate(N, *i*, *nnarray*)
16. **endfor**

Populate (N, i, narray) (fungsi untuk membangkitkan sampel sintetis. *)*

17. **while** *N* <> 0

```
18.     Pilih nomor random antar 1 dan k, sebut saja nn, langkah ini memilih
        satu dari k tetangga yang berdekatan dengan i.
19.     for aatr ← 1 to numattrs
20.         Hitung: dif = Sampel(nnarray[nn][aatr]-Sampel[i][aatr]
21.         Hitung: gap = nomor acak antara 0 sampai dengan 1
22.         Sintetis [newindex][aatr]=Sampel[i][aatr]+gap*dif
23.     endfor
24.     newindex++
25.     N = N - 1
26. endwhile
27. return(*End of populate*)
End of Pseudo-Code
```



Gambar 2.3 Ilustrasi dari *k-nearest neighbor*
(Sumber: Belur V. Dasarathy, *ed. 1991)*

2.7 *Bayesian Classification*

Bayesian classification merupakan metode pengklasifikasian statistik yang dapat digunakan untuk memprediksi peluang keanggotaan suatu kelas, seperti probabilitas dari sebuah *tuple* yang dimiliki oleh suatu kelas tertentu. *Bayesian classification* didasarkan pada teori Bayesian. Penelitian yang membandingkan algoritma klasifikasi menemukan bahwa klasifikasi Bayes sederhana yang disebut sebagai *naïve Bayesian classifier* mempunyai performa yang sebanding jika dibandingkan dengan *decision tree* dan beberapa metode *neural network*. *Naïve Bayesian classifiers* mengasumsikan bahwa pengaruh nilai atribut dari suatu kelas tidak berhubungan dengan nilai atribut yang lainnya (*class conditional independence*). *Bayesian classification* terbukti memiliki akurasi dan kecepatan yang tinggi saat

diaplikasikan ke dalam database dengan data yang besar (Han & Kamber, 2006).

Misalkan $y=(y_1,y_2,\ldots,y_n)^T$ adalah vektor dengan jumlah observasi sebanyak *n* yang mengikuti distribusi tertentu dengan fungsi densitas (PDF), $p(y|\theta)$, dimana $\theta=(\theta_1,\theta_2,\ldots,\theta_k)^T$ adalah sebuah vektor parameter dengan banyaknya kelas yang terbentuk sebanyak *k* dan memiliki probabilitas distribusi $p(\theta)$. Berdasarkan teorema Bayes, dapat ditentukan distribusi *posterior* dari θ, $p(\theta|y)$, sesuai dengan persamaan (2.4).

$$p(\theta|y)=\frac{p(y|\theta)p(\theta)}{p(y)} \tag{2.4}$$

Untuk menghitung $p(y|\theta)$ maka digunakan asumsi *class conditional independence*, maka dapat dirumuskan sebagai berikut.

$$\begin{aligned} P(y|\theta) &= \prod_{k=1}^{n} P(y_k|\theta) \\ &= P(y_1|\theta)\times P(y_2|\theta)\times\cdots\times P(y_n|\theta) \end{aligned} \tag{2.5}$$

Pada setiap atribut $y=(y_1,y_2,\ldots,y_n)^T$, dilihat terlebih dahulu apakah nilai atribut tersebut adalah kategorik atau bernilai kontinu. Dengan $p(y|\theta)$ merupakan fungsi *likelihood* yang berisi informasi sampel data, sedangkan $p(\theta)$ adalah fungsi distribusi *prior* dari θ dan $p(y)$ adalah fungsi konstanta densitas, dimana:

$$p(y)=\begin{cases} \int p(y|\theta)p(\theta)\,d\theta & \theta \text{ kontinu} \\ \sum p(y|\theta)p(\theta) & \theta \text{ diskrit} \end{cases} \tag{2.6}$$

Jika $y=(y_1,y_2,\ldots,y_n)^T$ bernilai kontinu, maka atribut yang bernilai kontinu diasumsikan memiliki distribusi Gaussian dengan mean μ dan standard deviasi σ yang didefinisikan sebagai berikut.

$$g\left(y\middle|\mu,\sigma^2\right)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (2.7)$$

Sehingga,

$$P\left(y|\theta\right)=g\left(y,\mu_\theta,\sigma_\theta^2\right) \quad (2.8)$$

Sehingga persamaan (2.4) dapat dinyatakan dalam proporsional sesuai dalam persamaan (2.9).

$$p(\theta|y)\propto p\left(y|\theta\right)p(\theta) \quad (2.9)$$

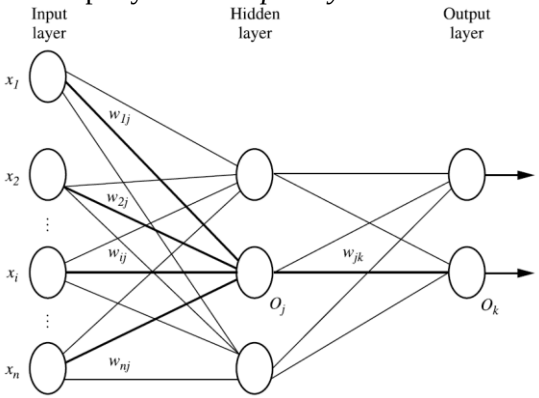
Persamaan (2.9) menunjukkan bahwa distribusi *posterior* merupakan kombinasi antara distribusi *prior* dan data observasi yang digunakan untuk membangun fungsi *likelihood.* Distribusi *prior* merupakan informasi awal yang dibutuhkan untuk membentuk distribusi *posterior*, selain juga dibutuhkan informasi dari sampel yang dinyatakan melalui *likelihood* (Box & Tiao, 1973).

2.8 *Artificial Neural Network*

Artificial Neural Network (ANN) atau bisa disebut sebagai Jaringan Syaraf Tiruan (JST) merupakan cabang dari ilmu kecerdasan buatan (*artificial intelligence*). Menurut Pujiadi dan Widyaiswara (2013), ANN merupakan salah satu sistem pemrosesan informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah. Pembuatan ANN terinspirasi dari kesadaran atas *complex learning system* pada otak yang terdiri dari *set-set* neuron yang saling berhubungan secara dekat (Meinanda et al., 2009). Dalam klasifikasi, ANN dapat membuat perkiraan yang cukup fleksibel, artinya ANN dapat digunakan untuk membentuk suatu model linier maupun non linier (Bar-Yam dalam Meinanda, 2009).

Neural network adalah kumpulan dari *input/output* yang terhubung dimana setiap sambungan mempunyai bobot. Jaringan tersebut menyesuaikan bobot sehingga dapat memprediksi label

kelas yang benar dari input tuples. Algoritma backpropagation menunjukkan pembelajaran *multilayer feed-forward neural network* yang terdiri dari lapisan input, satu atau lebih lapisan tersembunyi, dan lapisan output (Han & Kamber, 2006). Contoh dari *multilayer feed-forward neural network* ditunjukkan dalam Gambar 2.3 mempunyai dua *output layer*.



Gambar 2.3 A multi*layer* feed-forward neural network
(Sumber: Han & Kamber, 2006)

Backpropagation secara berulang memproses data *training*, membandingkan prediksi jaringan untuk setiap tuple dengan nilai target yang diketahui. Setiap data *training*, bobot diubah sehingga dapat meminimalkan MSE (*mean aquared error*) antara hasil klasifikasi dan nilai target sebenarnya. Modifikasi ini dibuat dengan arah "*backwards*" atau mundur dari lapisan *output* melalui setiap lapisan kemudian menuju pada lapisan tersembunyi yang pertama. Oleh karena itu dinamakan backpropagation. Iterasi akan berhenti ketika bobot sudah konvergen (Han & Kamber, 2006).

Misalkan $x = (x_1, x_2, \ldots, x_n)$ adalah *input layer* dengan jumlah variabel *x* sebanyak *n* dan $O = (O_1, O_2, \ldots, O_j)$ adalah *output layer* sebanyak *j*, sehingga langkah-langkah yang digunakan dalam backpropagation adalah sebagai berikut.

1. Inisialisasi bobot: Bobot dalam jaringan diinisialisasi ke angka acak kecil.

2. *Propagate the inputs forward*: Pertama, data *training* dijadikan lapisan *input* dalam jaringan. Untuk unit input j, output O_j akan sama dengan inputnya I_j dengan rumus sebagai berikut.

$$I_j = \sum_{i=1}^{n} w_{ij} O_i + \theta_j \tag{2.10}$$

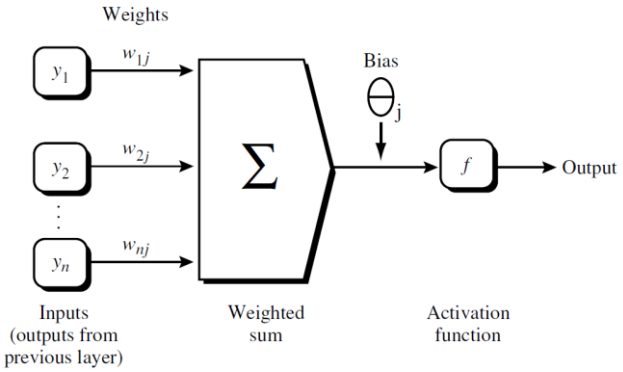
Keterangan :

w_{ij} = bobot penghubung dari unit ke-*i* pada lapisan sebelumnya ke unit *j*.

O_i = *output* dari unit *i* dari lapisan sebelumnya.

θ_j = bias dari unit. Bias bertindak sebagai permulaan yang berfungsi untuk memvariasikan aktivitas unit.

Setiap unit dalam *hidden layer* dan *output layer* mengambil *input layer* kemudian menerapkan fungsi aktivasi yang diilustrasikan pada Gambar 2.4.



Gambar 2.4 Ilustrasi Backpropagation (Sumber: Han & Kamber, 2006)

3. Menghitung *output* dari setiap unit j dengan rumus sebagai berikut.

$$O_j = \frac{1}{1 + e^{-I_j}} \tag{2.11}$$

4. *Backpropagate the error*: kesalahan disebarkan kebelakang dengan memperbarui bobot dan bias menggambarkan kesalahan prediksi jaringan dengan rumus sebagai berikut.

$$Err_j = O_j(1-O_j)(T_j - O_j) \quad (2.12)$$

Dimana O_j adalah nilai *output* yang sebenarnya dari unit ke *j*, dan T_j adalah nilai target yang diketahui dari data *training*. $O_j(1-O_j)$ adalah turunan dari fungsi logistik

5. Menghitung kesalahan sehubungan dengan lapisan setelahnya dengan rumus sebagai berikut.

$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk} \quad (2.13)$$

Dimana w_{jk} adalah bobot dan bias terbaru yang menggambarkan *propagated error*. Bobot terbaru menggunakan rumus sebagai berikut.

$$\Delta w_{ij} = (l) Err_j O_i \quad (2.14)$$

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (2.15)$$

Backpropagation menggunakan metode *gradient descent* untuk mencari satu set bobot yang sesuai dengan data *training* sehingga dapat meminimalkan jarak rata-rata kuadrat antara prediksi dan nilai target yang telah diketahui.

6. Menghitung selisih bias dengan rumus sebagai berikut.

$$\Delta\theta_j = (l) Err_j \quad (2.16)$$

$$\theta_j = \theta_j + \Delta\theta_j \quad (2.17)$$

(Han & Kamber, 2006).

2.9 Ketepatan Klasifikasi

Pengukuran ketepatan klasifikasi dilakukan untuk melihat performa klasifikasi yang telah dilakukan. Dalam mengukur ketepatan klasifikasi, perlu diketahui jumlah pada setiap kelas

prediksi dan kelas aktual yang terdiri dari *TP* (*True Positive*) yaitu jumlah *tweet* bersentimen positif yang tepat diprediksi dalam kelas positif, *TN* (*True Negative*) yaitu *tweet* yang tepat terprediksi dalam kelas negatif, *FP* (*False Positive*) yaitu *tweet* bersentimen negatif yang terprediksi dalam kelas positif, dan *FN* (*False Negative*) yaitu *tweet* bersentimen positif yang terprediksi dalam kelas negatif.

Tabel 2.1 Confusion Matrix

Kelas Aktual	**Kelas Prediksi**	
	Positif	Negatif
Positif	*TP*	*FN*
Negatif	*FP*	*TN*

Tabel 2.1 menunjukkan *confusion matrix* yang digunakan untuk menilai ketepatan klasifikasi. Nilai yang berada pada diagonal utama mewakili keputusan yang benar. Berikut adalah beberapa kriteria untuk menilai ketepatan klasifikasi.

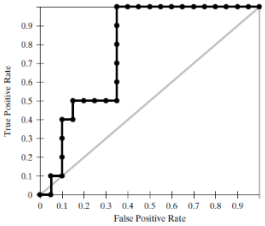
$$precision = \frac{TP}{TP + FP} \tag{2.18}$$

$$recall = \frac{TP}{TP + FN} \tag{2.19}$$

Receiver Operating Characterstics (ROC) adalah metode yang popular digunakan untuk menilai kinerja metode klasifikasi yang digunakan ketika terdapat dua kelas. ROC mensyaratkan bahwa metode klasifikasi mengeluarkan nilai skor kelas positif pada setiap titik untuk data *testing*. Skor ini kemudian digunakan untuk menyusun poin dalam urutan menurun. Sebagai contoh, probabilitas posterior $P(C_1|X_i)$ sebagai skor untuk penggolongan Bayes. Area di bawah kurva ROC (AUC) dapat digunakan sebagai ukuran kinerja metode klasifikasi. AUC terletak pada interval 0 sampai 1, semakin mendekati 1 maka nilai AUC akan semakin bagus. Nilai AUC adalah peluang metode klasifikasi akan mem

berikan peringkat *random positive test* case lebih tinggi dari *random negative test*.

Kurva ROC tidak peka terhadap *skew* kelas. Hal ini dikarenakan TPR (*True Positive Rates*) adalah peluang untuk memprediksi titik positif sebagai positif, dan FPS (*False Positif Rates*) adalah peluang untuk memprediksi titik negatif sebagai positif, tidak tergantung pada rasio ukuran kelas positif ke negatif. Kurva ROC pada dasarnya dapat digunakan untuk kelas yang seimbang atau *skewed* (ketika satu kelas memiliki lebih banyak poin dari pada yang lain) (Zaky & Meira, 2014).



Gambar 2.5 Contoh kurva ROC pada Iris principal components dataset. Kurva ROC untuk naïve Bayes (black) dan metode kalsifikasi yang diketahui (grey). (Sumber : Zaky & Meira, 2014)

2.10 *Social Network Analysis*

SNA memiliki beberapa definisi, diantaranya: Nooy, Mrvar, dan Batagelj (2005) mendefinisikan bahwa *Social Network Analysis* adalah proses pemetaan dan pengukuran relasi antara orang ke orang, sedangkan Freeman (1979) mendefinisikan sebagai teknik yang fokus mempelajari pola interaksi pada manusia yang tidak terlihat secara eksplisit. Scott (1992) mendefinisikan sebagai sekumpulan metode untuk menginvestigasi aspek relasi pada struktur sosial. Berdasarkan ketiga definisi tersebut, secara garis besar memiliki kesamaan makna, yaitu mengarah pada proses analisis jaringan sosial berkaitan dengan bentuk struktur dan pola interaksi entitas di dalamnya.

Sebagai contoh pada kasus jejaring sosial *online* di twitter, yang lebih banyak dianalisis adalah interaksi antar *user* twitter,

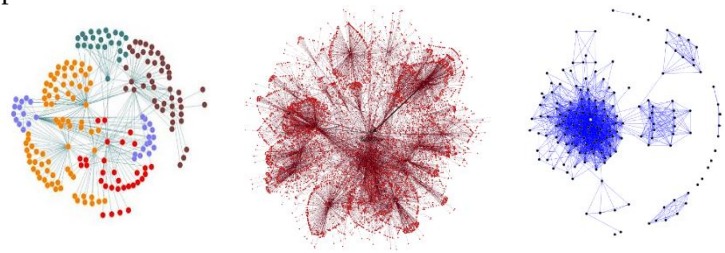
dalam hal ini pola interaksi akan dapat menentukan *user* mana yang paling berpengaruh dalam suatu lingkup grup tertentu. SNA dapat memetakan relasi antar orang, organisasi, topik, lokasi, dan intensitas informasi lainnya. Node atau titik di dalam jaringan menggambarkan orang, organisasi, lokasi, atau entitas informasi. Garis sambungan antar titik menggambarkan relasi antar titik. SNA didalam teori jaringan terdiri dari *node* dan *edge* (juga disebut relasi, link, atau koneksi). *Node* adalah seorang individu dalam jaringan, dan *edge* adalah hubungan antara *node*. Ada 2 macam graf yaitu *Graf Undirected* dan *Graf Directed*. *Graf Undirected* adalah graph yang hubungannya tidak mempunyai orientasi arah. Pada *graf undirected*, nilai antar node yang dihubungkan oleh edge tidak diperhatikan, yang penting saling berhubungan/berkoneksi maka memiliki nilai. *Graf directed* adalah graf yang setiap hubungan diberikan orientasi arah, dimana edgenya diperhatikan. Contoh dari *Graf Undirected* dan *Graf Directed* dapat ditunjukkan pada Gambar 2.6 dan Gambar 2.7.

Gambar 2.6 Gambaran *Graf Undirected* (Sumber: Cheliotis, 2010)

Gambar 2.7 Gambaran *Graf Directed* (Sumber: Cheliotis, 2010)

SNA dalam Twitter menampilkan peta relasi antar actor di sosial media. Retweet menandakan tentang persetujuan (*agreement*), sedangkan mention menandakan pertanyaan/diskusi (*discussion*). Ukuran (*metric*) yang digunakan dalam penentuan aktor dalam penelitian ini adalah *degree centrality*, *betweenness*

centrality, dan *closeness centrality*. Bentuk visualisasi dari SNA dapat dilihat dalam Gambar 2.8.



Gambar 2.8 Visualisasi Social Network Analysis
(Sumber: Wasserman & Faust, 1994)

2.10.1 *Degree Centrality*

Defgree centrality dapat didefinisikan sebagai ukuran kedekatan aktor utama yang paling aktif atau memiliki kedekatan paling banyak dengan actor di dalam suatu jaringan. Misalkan $(n_i = n_1, n_2, ..., n_g)$ adalah aktor dalam suatu jaringan, maka *Actor-level degree centrality index* adalah sebagai berikut (Wasserman & Faust, 1994).

$$C_D(n_i) = d(n_i) = x_{i+} = \sum_{j=1}^{g} x_{ij} \qquad (2.20)$$

Keterangan :

$C_D(n_i)$ = *actor-level degree centrality index* pada aktor ke-*i*.

$d(n_i)$ = banyaknya garis yang ada dalam jaringan

x_{ij} = kedekatan node ke *i* dan node ke *j*

Suatu *node* mempunyai nilai *degree centrality* antara 0 hingga *g-1*. Dimana *g* adalah aktor yang terdapat dalam suatu jaringan.

2.10.2*Closeness Centrality*

Ukuran kedekatan aktor yang kedua didasarkan pada kedekatan atau jarak yang disebut dengan *closeness centrality*. Metode ini focus pada seberapa dekat aktor utama terhadap aktor yang lainnya. Rumus yang digunakan adalah sebagai berikut (Wasserman & Faust, 1994).

$$C_c(n_i) = \left[\sum_{j=1}^{g} d(n_i, n_j) \right]^{-1} \qquad (2.21)$$

Keterangan :

$C_c(n_i)$ = *actor closeness index* pada aktor ke-*i*

$d(n_i, n_j)$ = jumlah garis pada actor *i* dan *j* dengan menggunakan *geodesic linking*.

Saat maksimal apabila indeks sama dengan $(g-1)^{-1}$, yang muncul ketika aktor berada berdekatan dengan semua aktor lainnya. Pada saat minimal, indeks mencapai nilai 0, yang muncul setiap satu atau lebih aktor tidak dapat dijangkau dari aktor utama. Sebuah *node* dikatakan dapat dijangkau dari yang lain apabila ada jalur yang menghubungkan dua node.

2.10.3 *Betweenness Centrality*

Interaksi antara dua aktor yang tidak berdekatan mungkin bergantung pada aktor lain. Terutama aktor yang berada di jalur antara keduanya. Aktor yang lain ini berpotensi memiliki pengendalian atas interaksi antara dua aktor yang tidak berdekatan. Indeks *betweennes centrality* adalah jumlah dari estimasi peluang atas semua pasangan aktor yang tidak termasuk aktor ke-i . Berikut adalah rumus untuk menghitung *betweennes centrality* (Wasserman & Faust, 1994).

$$\text{Group Betweeness Centrality} = \sum_{u<v} \frac{g_{u,v}(C)}{g_{u,v}} \qquad u, v \notin C \quad (2.22)$$

Keterangan :

g_{uv} = nilai *geodesic linking* yang menghubungkan antara node ke *u* dan *v*

$g_{uv}(C)$ = nilai *geodesic linking* yang menghubungkan antara node ke *u* dan *v* yang melewati node C.

2.11 *Word Cloud*

Word cloud merupakan salah satu metode visualisasi dokumen teks yang sering digunakan. *Word cloud* merupakan representasi grafis dari sebuah dokumen yang dilakukan dengan *plotting* kata-kata yang sering muncul pada sebuah dokumen pada ruang dua dimensi. Frekuensi dari kata yang sering muncul ditunjukkan melalui ukuran huruf kata tersebut. Semakin besar ukuran kata menunjukkan semakin besar frekuensi kata tersebut mundul dalam dokumen. Contoh dari visualisasi dokumen teks dengan *word cloud* ditunjukkan pada Gambar 2.3 (Castella & Sutton, 2014).

Gambar 2.9 Visualisasi Data dengan *Word Cloud*
(Sumber: worditout.com)

2.12 PT. Tokopedia

PT. Tokopedia merupakan salah satu mall *online* di Indonesia yang mengusung model bisnis *marketplace* dan *mall online*. Wujud sebuah mall *online* yang mempertemukan penjual dan pembeli dan memungkinkan untuk terjadinya transaksi jual

beli *online* dengan aman dan nyaman. Bergabung untuk menggunakan Tokopedia sangatlah mudah dan tidak dipungut biaya. Setelah beroperasi www.tokopedia.com telah menjadi salah satu *online marketplace* dengan tingkat pertumbuhan yang sangat pesat di Indonesia walaupun usianya masih seumur jagung, baik dalam jumlah anggota, took, *online* aktif, jumlah produk hinggajumlah transaksi pembelian dan penjualan setiap harinya. Tokopedia sudah mampu bersaing di pasar *marketplace* Indonesia, selain mempunyai metode yang berbeda dari pesaingnya Tokopedia mampu terus maju dalm persaingan bisnis *e-commerce*. Tokopedia sejatinya tidak mempunyai cabang perusahaan. Tokopedia hanya memiliki kantor pusat yang berlokasi di Jakarta namun memiliki berbagai pengguna (penjual) diseluruh penjuru Indonesia.

Gambar 2.10 Logo Tokopedia (Sumber: Tokopedia.com)

2.13 Bukalapak

Bukalapak merupakan salah satu *online marketplace* terkemuka di Indonesia yang menyediakan sarana jual–beli dari konsumen ke konsumen. Semua orang dapat membuka toko *online* di Bukalapak dan melayani pembeli dari seluruh Indonesia untuk transaksi satuan maupun banyak. Bukalapak memiliki slogan jual-beli *online* mudah dan terpercaya karena Bukalapak memberikan jaminan 100% uang kembali kepada pembeli jika barang tidak dikirimkan oleh pelapak.

Visi Bukalapak : Menjadi *online* marketplace nomor 1 di Indonesia

Misi Bukalapak : Memberdayakan UKM yang ada di seluruh penjuru Indonesia

Bukalapak sebagai sarana penunjang bisnis berusaha menyediakan berbagai fitur dan layanan untuk menjamin keamanan dan kenyamanan para penggunanya. Bukalapak tidak

berperan sebagai Pelapak barang, melainkan sebagai perantara antara Pelapak dan Pembeli, untuk mengamankan setiap transaksi yang berlangsung di dalam *platform* Bukalapak melalui mekanisme Bukalapak Payment System. Adanya biaya ekstra (termasuk pajak dan biaya lainnya) atas segala transaksi yang terjadi di Bukalapak berada di luar kewenangan Bukalapak sebagai perantara, dan akan diurus oleh pihak-pihak yang bersangkutan (baik Pelapak atau pun Pembeli) sesuai ketentuan yang berlaku di Indonesia.

Gambar 2.11 Logo Bukalapak (Sumber: Bukalapak.com)

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Sumber data yang digunakan dalam penelitian ini adalah sumber data primer yang diambil dari kumpulan *tweet* pengguna Twitter di Indonesia. Akun Twitter yang digunakan dalam analisis kali ini adalah @tokopediacare (*costumer care* dari Tokopedia) dan @bukabantuan (*costumer care* dari Bukalapak). Data tersebut diambil dari tanggal 6 Februari 2018 – 6 Maret 2018 dengan menggunakan Twitter API (*Application Programming Interface*).

3.2 Struktur Data dan Variabel Penelitian

Data yang digunakan dalam penelitian ini dibagi menjadi data training sebanyak 90% dan data testing sebanyak 10% menggunakan 10-*fold cross validation*. Variabel penelitian yang digunakan dalam penelitian ini diberikan dalam Tabel 3.1

Tabel 3.1 Variabel Penelitian

Variabel	Keterangan	Skala Data
Y	Sentiment (Positif/Negatif) 0 = Sentiment Positif 1 = Sentimen Negatif	Nominal
X_k	Kata Kunci (Frekuensi Kemunculan Kata ke-*k* pada tweet ke-*n*)	Rasio

Berdasarkan Tabel 3.1 dapat diketahui bahwa variabel respon (Y) yang digunakan adalah sentimen dari pelanggan Tokopedia dan Bukalapak, sedangkan variabel prediktor (X) adalah frekuensi kemunculan kata ke-*k* pada tweet ke *n*.

Struktur data tweet yang diambil dari akun @tokopediacare dan @bukabantuan diberikan dalam Tabel 3.2

Tabel 3.2 Struktur Data Penelitian Sebelum *Pre-Processing*

No.	Teks	*SN*	*Reply to SN*	*Created*	*Sentiment* (Positif/Negatif)
1	$Teks_1$	$Account_1$	$Account_1$	$Time_1$	$Sentiment_1$
2	$Teks_2$	$Account_2$	$Account_2$	$Time_2$	$Sentiment_2$
.	.	.	.	.	.
.	.	.	.	.	.
n	$Teks_n$	$Account_n$	$Account_n$	$Time_n$	$Sentiment_n$

Notasi *n* menunjukkan jumlah data yang digunakan dalam penelitian. Kolom SN (*Screen Name*) adalah akun yang membuat *tweet*, sedangkan kolom *Reply to SN* menunjukkan akun kepada siapa *tweet* tersebut ditujukan. Kolom *created* adalah menunjukkan waktu *tweet* tersebut dipublikasikan. Kolom *sentiment* adalah pendapat pelanggan Tokopedia maupun Bukalapak yang dibagi menjadi dua opini yaitu positif dan negatif. Setelah dilakukan *pre-processing* data, struktur data yang terbentuk untuk klasifikasi menggunakan metode *Naïve Bayes Clasifier* dan *Artificial Neural Network* adalah sebagai berikut

Tabel 3.3 Struktur Data Penelitian Setelah *Pre-Processing*

No.	*Screen Name*	*Tweet* (*t*)	Klasifikasi Sentimen (*y*)	Kata Kunci (x_1)	Kata Kunci (x_2)	...	Kata Kunci (x_k)
1	$account_1$	t_1	y_1	$x_{1,1}$	$x_{1,2}$	...	$x_{1,k}$
2	$account_2$	t_2	y_2	$x_{2,1}$	$x_{2,2}$	...	$x_{2,k}$
.	.	.	.	.	.	...	.
.	.	.	.	.	.	...	.
n	$account_n$	t_n	y_n	$x_{n,1}$	$x_{n,2}$	...	$x_{n,k}$

3.3 Langkah Analisis

Langkah analisis yang digunakan dalam dalam penelitian ini adalah sebagai berikut.

1. Mengambil data *tweet* dengan menggunakan Twitter API.
   a. Memasukkan *keyword* @Tokopediacare dan @bukabantuan.
   b. Mengatur waktu *tweet* tersebut dibuat.

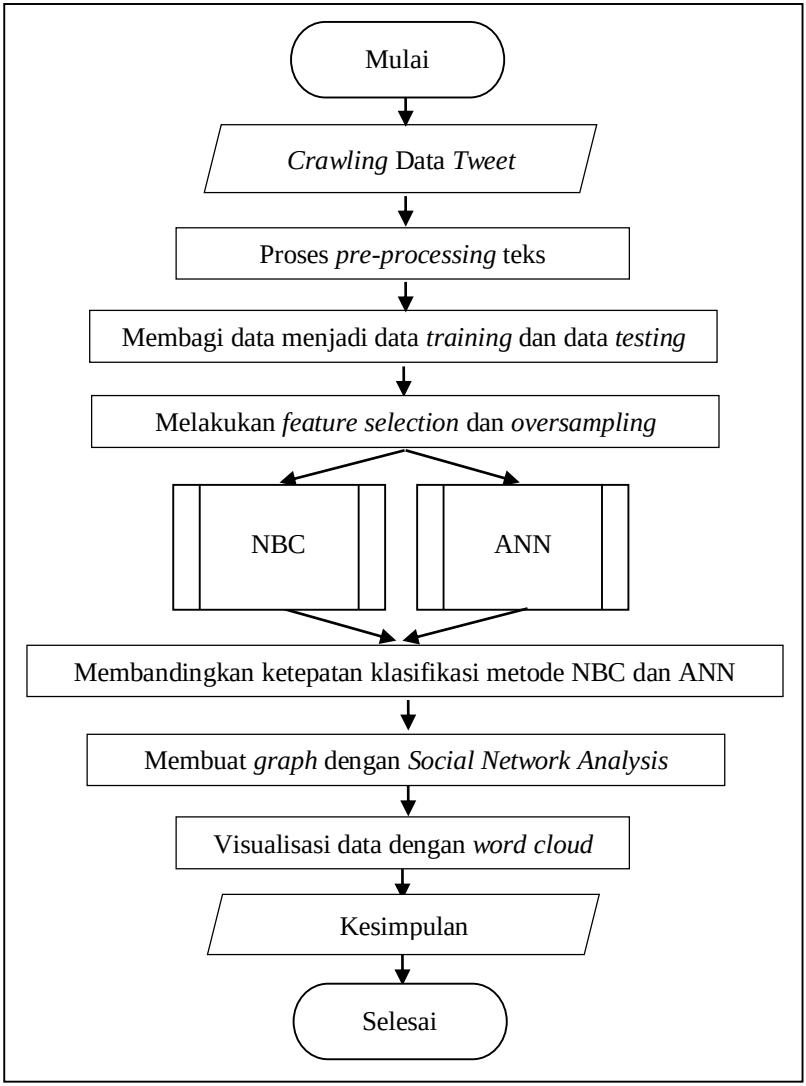
   c. Menyimpan hasil *crawling data* ke *database*.
2. Membagi data menjadi data *training* dan *testing* yaitu dengan perbandingan 90:10 dengan menggunakan metode 10-*fold cross validation.*
3. Melakukan *pre-processing* dokumen teks yang meliputi proses *stemming*, *filtering stopwords*, *case folding*, dan *tokenizing*. Daftar *stopwords* diambil dari thesis yang ditulis oleh F. Z. Tala yang berjudul "*A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*" (Tala, 2003) sedangkan kata dasar diambil dari Kamus Besar Bahasa Indonesia.
4. Melakukan memilihan variabel dengan menggunakan *feature selection* χ^2
5. Melakukan oversampling dengan metode SMOTE (*Synthetic Minority Oversampling Technique*)
6. Melakukan klasifikasi menggunakan metode *Naïve Bayes Clasifier*.
   a. Menghitung probabilitas V_j pada data *training*, dimana V_j merupakan kategori sentimen, yaitu V_1 = negatif dan V_2 = positif.
   b. Menghitung probabilitas kata a_i pada kategori V_j
   c. Menghitung probabilitas *Naïve Bayes Clasifier* disimpan dan digunakan untuk tahap data *testing*.
   d. Menghitung probabilitas tertinggi dari kategori sentimen yang diujikan (V_{MAP})
   e. Mencari nilai V_{MAP} paling maksimum dan memasukkan *tweet* tersebut pada kategori dengan V_{MAP} maksimum.
   f. Menghitung nilai akurasi dari model yang terbentuk.
7. Melakukan klasifikasi menggunakan metode *Artificial Neural Network*.
   a. Menentukan besar inisiasi bobot yang dicoba sebanyak *l* kali.
   b. Menentukan inisiasi bobot
   c. Mempropagasikan data input ke depan

d. Menghitung *input* unit ke-j dengan memperhatikan lapisan sebelumnya dan menghitung *output* setiap unit ke-j.
e. Menghitung *error* pada lapisan *output* dan lapisan tersembunyi.
f. Menghitung koreksi bobot
g. Menghitung peningkatan dan pembaharuan bias.
h. Melakukan iterasi sebanyak *l* kali dengan mengulangi langkah 5b-5g
i. Memilih inisiasi bobot yang menghasilkan solusi optimum.

8. Membandingkan hasil klasifikasi menggunakan metode *Naïve Bayes Clasifier* dan *Artificial Neural Network*.
9. Membuat graf pola interaksi dengan menggunakan metode *Social Network Analysis.*
10. Membuat visualisasi data teks twitter dengan membuat *plotting* kata-kata yang sering muncul dengan *word cloud*.

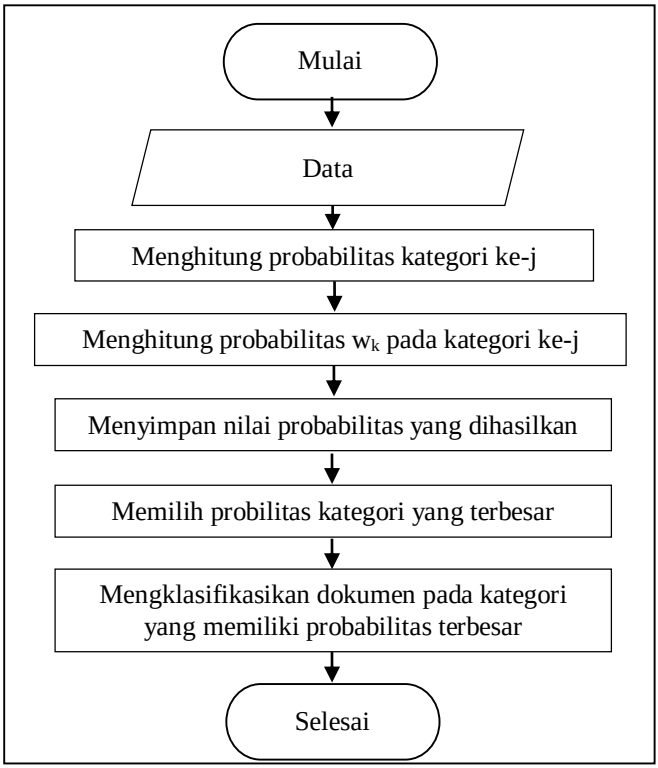
3.4 Diagram Alir

Langkah-langkah analisis dalam Sub Bab 3.3 diberikan dalam bentuk diagram alir pada Gambar 3.1, Gambar 3.2 dan Gambar 3.3



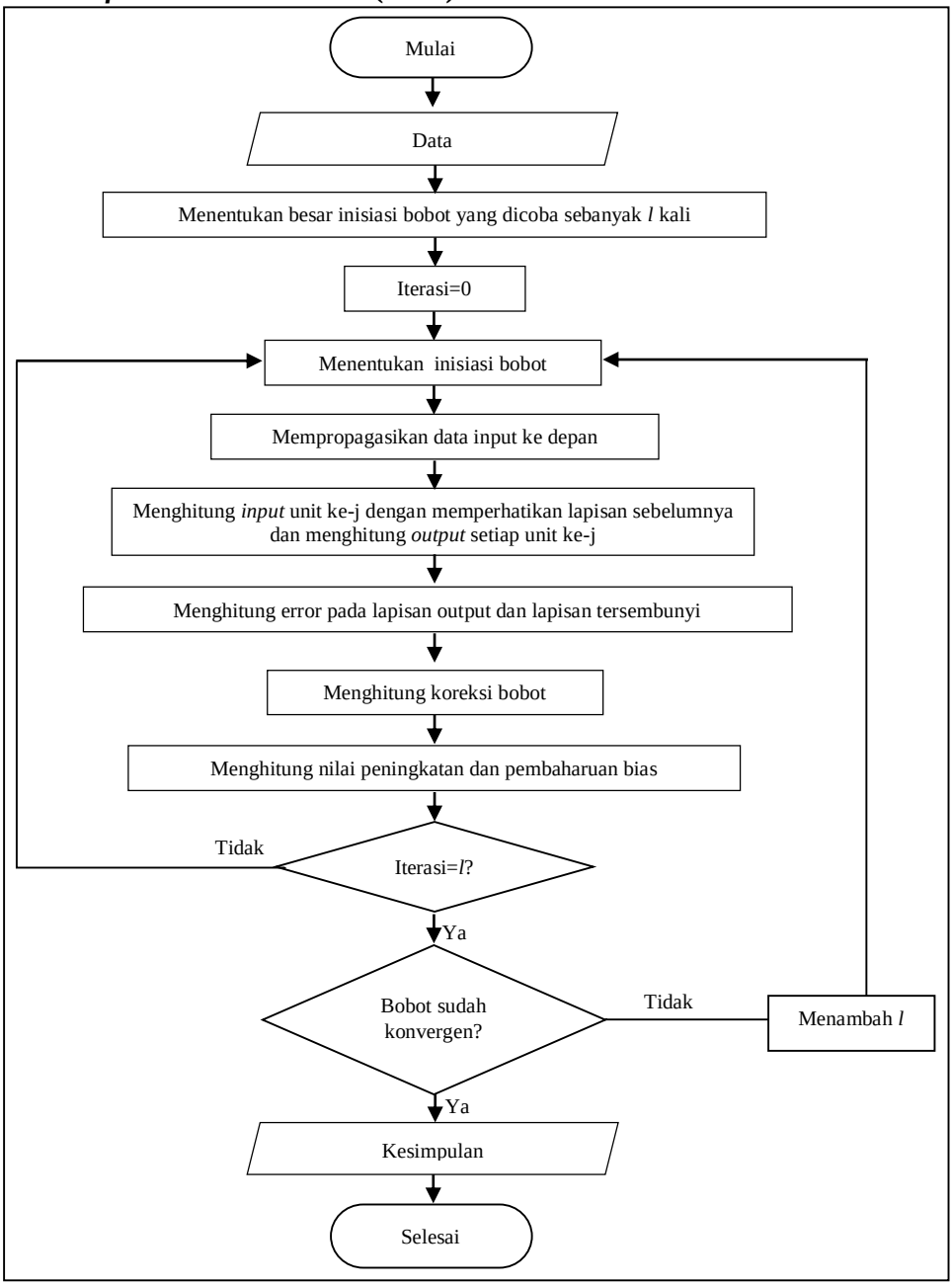
Gambar 3.1 Diagram Alir Penelitian

NBC (*Naïve Bayes Classifier*)



Gambar 3.2 Diagram Alir *Naïve Bayes Classifier*

***Artificial Neural Network* (ANN)**



Gambar 3.3 Diagram Alir *Artificial Neural Network*

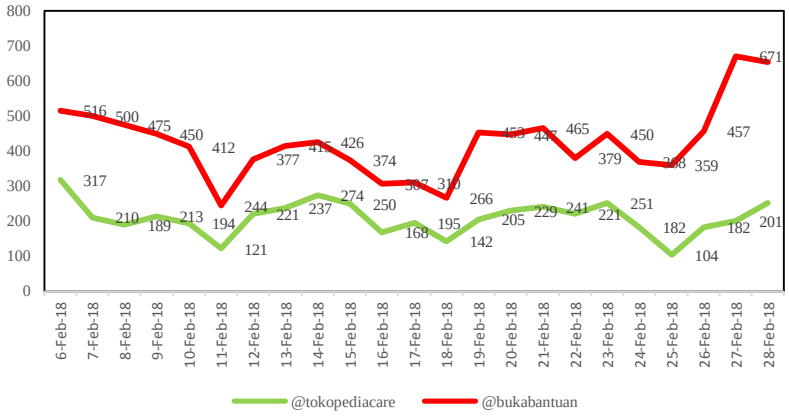
(halaman ini sengaja dikosongkan)

BAB IV
ANALISIS DAN PEMBAHASAN

Pada bab ini akan dibahas hasil analisis berdasarkan pengolahan data pada akun twitter @tokopediacare dan @bukabantuan yang telah dilakukan. Metode yang digunakan dalam klasifikasi teks twitter adalah *Naïve Bayes Classifier* (NBC) dan *Artificial Neural Network* (ANN), sedangkan metode yang digunakan untuk melihat pola jaringan konsumen adalah *Social Network Analysis* (SNA)

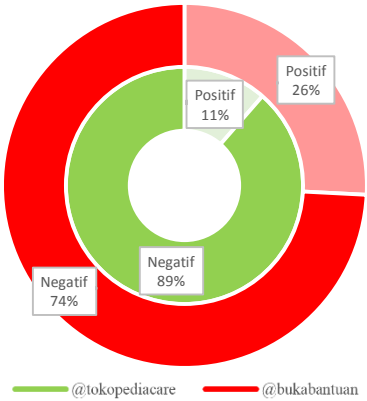
4.1 Karakteristik Data dan Praproses Data

Akun twitter @tokopediacare dan @bukabantuan adalah salah satu bentuk layanan pelanggan khusus melalui *online* yang disediakan untuk menanggapi tanggapan, pendapat, kritik, saran dan masalah *complain*. Secara manual, dilakukan pengumpulan data terhadap jumlah *post* dari konsumen yang ditujukan kepada akun twitter @tokopediacare dan @bukabantuan. Gambar 4.1 menunjukkan jumlah tweet dari konsumen pada bulan Februari 2018.



Gambar 4.1 Tren Jumlah Tweet Konsumen

Gambar 4.1 menunjukkan jumlah tweet konsumen pada bulan Februari 2018. Konsumen yang memberikan tanggapan, pendapat, kritik, saran dan masalah *complain* lebih banyak ditujukan pada akun @bukabantuan dari pada akun @tokopediacare. Jumlah tweet yang ditujukan pada kedua akun pada bulan Februari 2018 cenderung stabil, namun sedikit terjadi kenaikan pada akun @bukabantuan di akhir bulan februari. Hal tersebut dapat diakibatkan oleh beberapa faktor, salah satunya contohnya adalah kecenderungan orang akan berberlanja pada akhir bulan sehingga menyebabkan penjualan belanja *online* meningkat pada akhir bulan.



Gambar 4.2 Prosentase Sentimen Positif dan Negatif

Berdasarkan Gambar 4.2 dapat diketahui bahwa data yang digunakan untuk klasifikasi terdiri dari sentimen negatif dan positif. Sentimen negatif pada kedua akun @tokopediacare dan @bukabantuan memiliki prosentase yang lebih bayak jika dibandingkan dengan prosentase sentimen positif. Prosentase sentimen negatif pada akun @tokopediacare sebesar 89%, sedangkan prosentase sentimen positif adalah sebesar 11%.

Prosentase negatif pada akun @bukabantuan sebesar 74%, sedangkan prosentase sentimen positif sebesar 26%.

Sebelum dilakukan analisis, langkah selanjutnya yang akan dilakukan adalah pra proses pada data twitter @tokopediacare dan @bukabantuan terdiri dari beberapa tahapan, yaitu *case folding, cleansing, stopword, stemming,* dan *tokenizing.* Proses *case folding* adalah mengubah data teks menjadi *lowercase* dengan tujuan agar kata yang sama namun berbeda secara penulisan huruf kapital dan tidak, tidak dianggap kata yang berbeda. Contoh data sebelum dan sesudah melalui tahap *lowercase* ditampilkan pada Tabel 3.1

Tabel 4.1 Contoh Data Sebelum dan Sesudah Melalui Tahap *Lowercase*

Sebelum *Lowercase*	**Setelah *Lowercase***
@tokopediacare Barang yang dikirim tidak sesuai dengan yang dijanjikan, seller mengatakan barang ORIGINAL, tapi setelah saya cek ternyata palsu… https://t.co/nosy9l67pU.	@tokopediacare barang yang dikirim tidak sesuai dengan yang dijanjikan, seller mengatakan barang original, tapi setelah saya cek ternyata palsu… https://t.co/nosy9l67pu.
⋮	⋮
@BukaBantuan Baik. Barusan sudah ada jawaban, dana kelebihan pembayaran sudah dikembalikan ke BukaDompet. Terima kasih untuk bantuannya…	@bukabantuan baik. barusan sudah ada jawaban, dana kelebihan pembayaran sudah dikembalikan ke bukadompet. terima kasih untuk bantuannya…

Setelah melakukan proses *case folding* pada data teks, langkah selanjutnya adalah proses *cleansing*. *Cleansing* pada data twitter merupakan proses membersihkan *tweet* dari kata yang tidak diperlukan untuk mengurangi *noise*. Kata yang dihilangkan dalam twitter adalah karakter HTML, *emoticons*, *hastag (#)*, *username (@username)*, dan *URL*. Contoh data sebelum dan sesudah melalui tahap *cleansing* ditampilkan pada Tabel 4.2.

Tabel 4.2 Contoh Data Sebelum dan Sesudah Melalui Tahap *Cleansing*

Sebelum *Cleansing*	Setelah *Cleansing*
@tokopediacare barang yang dikirim tidak sesuai dengan yang dijanjikan, seller mengatakan barang original, tapi setelah saya cek ternyata palsu… https://t.co/nosy9l67pu.	barang yang dikirim tidak sesuai dengan yang dijan-jikan, seller mengatakan barang original, tapi setelah saya cek ternyata palsu
⋮	⋮
@bukabantuan baik. barusan sudah ada jawaban, dana kelebihan pembayaran sudah dikembalikan ke bukadompet. terima kasih untuk bantuannya…	baik. barusan sudah ada jawaban, dana kelebihan pembayaran sudah dikemba-likan ke bukadompet. terima kasih untuk bantuannya...

Proses *stemming* dilakukan untuk mendapatkan kata dasar dengan cara menghilangkan awalan, akhiran, sisipan, dan *confixes* (kombinasi dari awalan dan akhiran). Pada penelitian ini algoritma *stemming* yang digunakan adalah *Confix Striping Stemmer* yang merupakan pengembangan dari algoritma *Nazief and Adriani's Stemmer*. Berikut adalah contoh data sebelum dan sesudah melalui tahap *stemming* ditampilkan pada Tabel 4.3

Tabel 4.3 Contoh Data Sebelum dan Sesudah Melalui Tahap *Stemming*

Sebelum *Stemming*	Setelah *Stemming*
barang yang dikirim tidak sesuai dengan yang dijan-jikan, seller mengatakan barang original, tapi setelah saya cek ternyata palsu	barang yang kirim tidak sesuai dengan yang janji seller kata barang original tapi telah saya cek nyata palsu
⋮	⋮

Tabel 4.3 Contoh Data Sebelum dan Sesudah Melalui Tahap *Stemming (lanjutan)*

baik. barusan sudah ada jawaban, dana kelebihan pembayaran sudah dikembalikan ke bukadompet. terima kasih untuk bantuannya...	baik barusan sudah ada jawab dana lebih bayar sudah kembali ke bukadompet terima kasih untuk bantu

Langkah selanjutnya adalah *stopword* dan *tokenizing*. *Stopword* bekerja dengan menghilangkan kata-kata yang tidak menyampaikan pesan apapun secara signifikan pada suatu teks. Daftar *stopwords* yang digunakan adalah *stopwords* Bahasa Indonesia yang disusun berdasarkan penelitian Fadillah Z Tala pada tahun 2003. Sedangkan *tokenizing* adalah proses memcah kalimat menjadi kata-kata. Contoh data sebelum dan sesudah melalui tahap *stopword* dan *tokenizing* ditampilkan pada Tabel 4.4.

Tabel 4.4 Contoh Data Sebelum dan Sesudah Melalui Tahap *Stopword dan Tokenizing*

Sebelum *Stopword* dan *Tokenizing*	**Setelah *Stopword* dan *Tokenizing***
barang yang kirim tidak sesuai dengan yang janji seller kata barang original tapi telah saya cek nyata palsu	'barang', 'kirim', 'sesuai', 'janji', 'seller', 'barang', 'original', 'palsu'
⋮	⋮
baik barusan sudah ada jawab dana lebih bayar sudah kembali ke bukadompet terima kasih untuk bantu	'barusan', 'jawab', 'dana', 'bayar', 'bukadompet', 'bantu'

Hasil dari proses *stopword* dan *tokenizing* dipakai sebagai kata kunci dari data tweet yang akan dianalisis dengan menggunakan metode NBC dan ANN. Tabel 4.5 menunjukkan

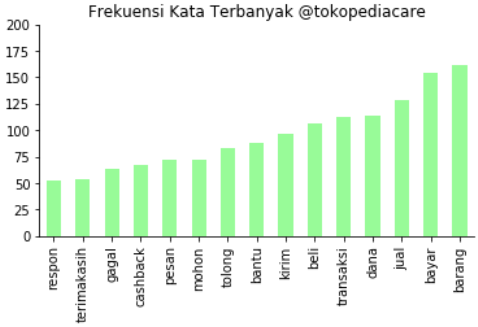
perhitungan frekuensi tiap kata kunci pada akun @tokopediacare dan @bukabantuan.

Tabel 4.5 *Count Vectorizer* pada data tweet

Tweet	Sentimen	Kata Kunci								
		admin	agen	air	akun	…	upload	valid	warna	web
1	0	0	1	0	0	…	1	0	1	0
2	1	1	0	0	1		0	0	0	0
3	0	0	0	0	0		0	1	1	0
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	1	1	0	0	1	…	0	0	0	0

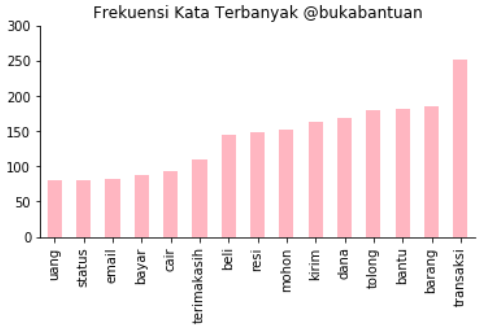
Berdasarkan Tabel 4.5 menunjukkan perhitungan frekuensi pada kata kunci. Pada tweet pertama didapatkan bahwa kata "agen" disebutkan pada sebanyak 1 kali, tetapi tidak disebutkan pada tweet ke-n, sedangkan kata "admin" disebutkan sebanyak 1 kali pada tweet ke-2, tetapi tidak disebutkan pada tweet pertama, dan seterusnya.

Setelah dilakukan *pre-processing* pada data teks, jumlah kata kunci dapat diketahui pada akun twitter @tokopediacare dan @bukabantuan. Frekuensi setiap kata dalam tweet @tokopediacare dan @bukabantuan ditunjukan pada Gambar 4.3 dan Gambar 4.4



Gambar 4.3 Frekuensi Kata pada Akun @tokopediacare

Berdasarkan Gambar 4.3 dapat diketahui frekuensi dari 15 kata tertinggi akun @tokopediacare. Kata "barang" merupakan kata yang paling banyak dibicarakan oleh konsumen. Hal tersebut dikarenakan bahwa PT. Tokopedia adalah *market place* yang banyak menjual barang daripada jasa. Kata "bayar" merupakan kata kedua yang paling banyak dibicarakan, dikarenakan sistem jual-beli dari *market place* PT. Tokopedia menggunakan e-*money*.



Gambar 4.4 Frekuensi Kata pada Akun @bukabantuan

Berdasarkan Gambar 4.4 dapat diketahui frekuensi dari 15 kata tertinggi akun @bukabantuan. Kata "transaksi" merupakan kata yang paling banyak dibicarakan oleh konsumen Bukapakak, sedangkan kata "barang" merupakan kata yang paling banyak kedua dibicarakan oleh konsumen. Hal tersebut dikarenakan bahwa Bukalapak adalah *market place* yang banyak menjual barang daripada jasa.

4.2 *Feature Selection*

Feature Selection dalam *machine learning* disebut juga *variable selection* (seleksi variabel) adalah proses pemilihan veriabel yang relevan untuk digunakan dalam pembentukan model. Ada berbagai keuntungan dari *feature selection* antara lain dapat menyederhanaan model sehingga lebih mudah di interpretasi oleh peneliti/pembaca dan waktu pelatihan yang lebih singkat. Pada

penelitian ini, *feature selection* yang digunakan adalah χ^2 dengan hipotesis H_0, tidak ada hubungan antara variabel respon (Y) dan variabel prediktor (x_i) dan hipotesis H_1 ada hubungan antara variabel respon (Y) dan variabel prediktor (x_i). Keputusan H_0 akan ditolak jika nilai χ^2 lebih besar dari $\chi^2_{(0.05;1)}$ yaitu sebesar 3.841 dan P-*value* kurang dari taraf signifikan $\alpha = 0.05$. Hasil yang diperoleh pada Tabel 4.6 dan Tabel 4.7 merupakan nilai χ^2 pada setiap variabel prediktor akun @tokopediacare dan @bukabantuan.

Tabel 4.6 Nilai χ^2 pada variabel X di @tokopediacare

Variabel x_i	**Nilai *Chisquare***	**P-*Value***	**Keputusan Berdasarkan Nilai *Chisquare***	**Keputusan Berdasarkan Nilai P-*Value***
1	0.1101	0.7401	Gagal Tolak H_0	Gagal Tolak H_0
2	0.0530	0.8179	Gagal Tolak H_0	Gagal Tolak H_0
3	0.0831	0.7731	Gagal Tolak H_0	Gagal Tolak H_0
4	0.1468	0.7016	Gagal Tolak H_0	Gagal Tolak H_0
5	0.3655	0.5455	Gagal Tolak H_0	Gagal Tolak H_0
6	0.1055	0.7454	Gagal Tolak H_0	Gagal Tolak H_0
7	0.0591	0.8079	Gagal Tolak H_0	Gagal Tolak H_0
8	0.0469	0.8286	Gagal Tolak H_0	Gagal Tolak H_0
⋮	⋮	⋮	⋮	⋮
721	198.5428	0.0000	Tolak H_0	Tolak H_0
⋮	⋮	⋮	⋮	⋮
914	65.9876	0.0000	Tolak H_0	Tolak H_0
⋮	⋮	⋮	⋮	⋮
1745	0.0845	0.7713	Gagal Tolak H_0	Gagal Tolak H_0
1746	2.7809	0.0954	Gagal Tolak H_0	Gagal Tolak H_0

Berdasarkan hasil pada Tabel 4.6 dapat diketahui nilai χ^2 dan P-*value* pada setiap variabel prediktor. Variabel ke-721 adalah variabel yang mempunyai nilai χ^2 paling tinggi yaitu sebesar 198.5428 dan P-*value* adalah sebesar 0.0000. Keputusan yang

dapat diambil adalah Tolak H_0 yang berarti variabel ke-721 merupakan variabel yang berpengaruh terhadap hasil sentimen. Sebelum melakukan *feature selection*, nilai χ^2 diurutkan mulai dari nilai tertinggi hingga nilai terendah, kemudian dilakukan *feature selection* dan memilih 1500 kata dan 500 kata dengan urutan paling tinggi.

Tabel 4.7 Nilai χ^2 pada variabel X di @bukabantuan

Variabel x_i	**Nilai *Chisquare***	**P-*Value***	**Keputusan Berdasarkan Nilai Chisquare**	**Keputusan Berdasarkan Nilai P-Value**
1	0.1296	0.7188	Gagal Tolak H_0	Gagal Tolak H_0
2	0.2489	0.6178	Gagal Tolak H_0	Gagal Tolak H_0
3	0.1393	0.7089	Gagal Tolak H_0	Gagal Tolak H_0
4	0.1661	0.6835	Gagal Tolak H_0	Gagal Tolak H_0
5	0.1908	0.6622	Gagal Tolak H_0	Gagal Tolak H_0
6	0.1751	0.6755	Gagal Tolak H_0	Gagal Tolak H_0
7	0.2430	0.6220	Gagal Tolak H_0	Gagal Tolak H_0
8	0.5183	0.4715	Gagal Tolak H_0	Gagal Tolak H_0
⋮	⋮	⋮	⋮	⋮
913	350.4135	0.000	Tolak H_0	Tolak H_0
⋮	⋮	⋮	⋮	⋮
1887	242.5949	0.000	Tolak H_0	Tolak H_0
⋮	⋮	⋮	⋮	⋮
2132	0.1360	0.99659	Gagal Tolak H_0	Gagal Tolak H_0
2133	0.1389	0.996801	Gagal Tolak H_0	Gagal Tolak H_0

Berdasarkan Tabel 4.7 dapat diketahui dapat diketahui nilai χ^2 pada setiap variabel x yang mempengaruhi sentimen pada akun @bukabantuan. Variabel ke-913 merupakan variabel dengan nilai χ^2 yang paling besar, yaitu sebesar 350,4135 dan P-*value* sebesar 0.000. Keputusan yang didapatkan adalah Tolak H_0 dikarenakan nilai χ^2 lebih besar dari $\chi^2_{(0.05;1)}$ (3.841), sehingga

kesimpulan yang didapatkan adalah terdapat pengaruh antara variabel ke-913 terhadap sentimen pada akun @bukabantuan. Sebelum melakukan *feature selection*, nilai χ^2 diurutkan mulai dari nilai tertinggi hingga nilai terendah, kemudian dilakukan *feature selection* dan memilih 1500 kata dan 500 kata dengan urutan paling tinggi.

4.3 SMOTE (*Synthetic Minority Oversampling TEchnique*)

Berdasarkan karakteristik data, dapat diketahui bahwa sentimen negatif pada kedua akun @tokopediacare dan @bukabantuan memiliki prosentase yang lebih bayak jika dibandingkan dengan prosentase sentimen positif. Hal tersebut merupakan salah satu kondisi yang disebut dengan *imbalanced data*. *Imbalanced data* merupakan kondisi data dengan jumlah data suatu kelas melebihi jumlah data kelas yang lain. Kelas data mayoritas adalah sentimen negatif, sedangkan kelas data minoritas adalah sentimen positif. Hal ini merupakan masalah dalam klasifikasi data karena kebanyakan *classifier* cenderung memprediksi kelas mayor dan mengabaikan kelas minor sehingga akurasi kelas minor sangat kecil. Sebelum dilakukan analisis SMOTE, terlebih dahulu data dibagi menjadi data *training* dan data *testing*. Perbandingan data training dan data testing pada penelitian ini adalah 90:10. Jumlah keseluruhan data sentimen pada akun twitter @tokopediacare dan @bukabantuan dapat ditunjukkan pada Tabel 4.8.

Tabel 4.8 Jumlah Data Sentimen

Data	Training		Testing	
	Negatif	**Positif**	**Negatif**	**Positif**
@tokopediacare	1115	117	109	28
@bukabantuan	1680	575	179	72

Synthetic Minority Oversampling TEchnique (SMOTE) digunakan untuk menambah jumlah kelas positif menggunakan

teknik penarikan sampel secara acak sehingga jumlah klas positif berjumlah sama dengan klas negatif. Jumlah sentimen pada data training akun @tokopediacare terbagi menjadi 1115 sentimen negatif dan 117 sentimen positif, sedangkan data training akun @bukabantuan terbagi menjadi 1680 sentimen negatif dan 575 sentimen positif. Sedangkan jumlah data testing pada akun @tokopediacare adalah sebesar 109 untuk sentimen negatif dan 28 untuk sentimen positif. Pada @bukabantuan, data testing yang diujikan berjumlah 179 sentimen negatif dan 72 sentimen positif.

Tabel 4.9 menunjukkan prosentase sentimen positif dan negatif sebelum dan sesudah *oversampling*.

Tabel 4.9 Jumlah Sentimen (Y) Data *Training*

Data	Original		SMOTE	
	Negatif	**Positif**	**Negatif**	**Positif**
@tokopediacare	1115	117	1115	1115
@bukabantuan	1680	575	1680	1680

Berdasarkan Tabel 4.9 dapat diketahui bahwa perbandingan dari jumlah sentimen positif dan sentimen negatif pada data *training* dengan SMOTE adalah 50:50. sehingga dapat dikatakan bahwa data training telah seimbang untuk dilakukan pemodelan dengan metode klasifikasi *Naïve Bayes* dan *Artificial Neural Network*.

4.4 Klasifikasi Data Tweet Menggunakan *Naïve Bayes Classifier* (NBC)

Bayesian classification merupakan metode pengklasifikasian statistik yang dapat digunakan untuk memprediksi peluang keanggotaan suatu kelas. Langkah pertama dalam mengklasifikasikan data *tweet* adalah melatih model menggunakan data *training.* Model yang telah dilatih dengan data *training* kemudian digunakan untuk mengklasifikasikan data *testing* ke dalam dua kelas sentimen yaitu positif dan negatif.

4.4.1 Klasifikasi Data Tweet Menggunakan *Naïve Bayes Classifier* (NBC) pada Akun @tokopediacare

Sebelum dilakukan klasifikasi, dilakukan *feature selection/variabel selection. Feature selection* adalah sebuah proses memilih kata (variabel prediktor) yang digunakan untuk pemodelan. Keuntungan menggunakan *feature selection* sebelum melakukan pengolahan data lebih lanjut adalah mengurangi *overfitting*, meningkatkan akurasi, dan mengefisienkan waktu. Berikut adalah langkah untuk memprediksi suatu data *tweet* masuk ke dalam sentimen positif (1) atau negatif (0) pada data sebelum dilakukan *feature selection*.

Untuk menentukan suatu tweet masuk ke dalam klas negatif atau kelas positif dapat dilihat dari probabilitas $P(\mathrm{X}|C_i)P(C_i)$ berikut.

Tabel 4.10 Probabilitas Klasifikasi NBC pada Tokopedia

Testing **Tweet**	**Probabilitas Negatif**	**Probabilitas Positif**	**Keputusan**
1	9.999×10^{-01}	3.968×10^{-05}	Negatif
2	9.999×10^{-01}	5.580×10^{-05}	Negatif
3	9.999×10^{-01}	6.880×10^{-06}	Negatif
4	9.999×10^{-01}	1.082×10^{-05}	Negatif
5	9.816×10^{-01}	1.838×10^{-02}	Negatif
6	9.999×10^{-01}	9.224×10^{-07}	Negatif
7	9.999×10^{-01}	1.122×10^{-06}	Negatif
8	9.999×10^{-01}	1.882×10^{-08}	Negatif
9	9.999×10^{-01}	8.416×10^{-07}	Negatif
10	4.382×10^{-01}	9.995×10^{-01}	Positif
11	9.998×10^{-01}	1.312×10^{-04}	Negatif
12	9.994×10^{-01}	5.191×10^{-04}	Negatif
13	9.999×10^{-01}	3.517×10^{-08}	Negatif
14	9.997×10^{-01}	2.889×10^{-04}	Negatif
15	9.999×10^{-01}	5.346×10^{-05}	Negatif
⋮	⋮	⋮	⋮
136	4.166×10^{-01}	5.833×10^{-01}	Negatif

Berdasarkan Tabel 4.10 dapat diketahui probabilitas klasifikasi dengan metode Naïve Bayes Classifier (NBC) pada akun @tokopediacare. Jika probabilitas sentimen negatif lebih tinggi jika dibandingkan dengan probabilitas sentimen positif, maka akan memberikan hasil keputusan negatif begitupula sebaliknya. Pada data tweet yang pertama, probabilitas sentimen negatif adalah sebesar 0.999960319, sedangkan probabilitas sentimen positif adalah sebesar 0.000039681. Keputusan yang dapat diambil adalah tweet pertama masuk kedalam sentimen negatif, dikarenakan probabilitas sentimen negatif lebih besar jika dibandingkan dengan probabilitas sentimen positif.

Berdasarkan hasil klasifikasi pada Tabel 4.10 dapat diketahui nilai ketepatan klasifikasinya dengan membandingkan Y(prediksi) dan Y(aktual) yang ditunjukkan melalui *confusion matrix* pada Tabel 4.11

Tabel 4.11 *Confusion Matrix* @tokopediacare

Kelas Aktual	**Kelas Prediksi**	
	Negatif	Positif
Negatif	109	0
Positif	25	3

Berdasarkan Tabel 4.11 dapat diketahui bahwa terdapat jumlah sentimen negatif yang dipredisikan negatif berjumlah 109, sentimen negatif yang dipredisikan negatif berjumlah 0, sentimen positif yang diprediksikan negatif berjumlah 25, dan sentimen positif yang diprediksikan positif berjumlah 3. Berdasarkan hasil *confusion matrix* tersebut didapatkan nilai akurasi sebesar 0.90, presisi sebesar 0.88, *recall* sebesar 0.90.

Evaluasi dari kinerja model klasifikasi dapat juga dihitung dengan menggunakan *Receiver Operating Characteristics* (ROC) dengan mengkombinasikan sensitifitas dan spesifisitas. Area di bawah kurva ROC yang disebut AUC (*Area Under Curve*) dapat digunakan sebagai ukuran kinerja metode klasifikasi. AUC terletak pada interval 0 sampai 1, semakin mendekati 1 maka nilai AUC akan semakin bagus. Berdasarkan Tabel 4.9 dapat diketahui nilai

Area Under Curve (AUC) adalah sebesar 0.5536 sehingga dapat diketahui bahwa hasil klasifikasi tersebut belum cukup baik untuk klasifikasi sehingga perlu dilakukan oversampling dengan metode SMOTE (*Synthetic Minority Oversampling TEchnique*).

Langkah selanjutnya adalah membandingkan hasil tingkat akurasi dengan membagi data *training* dan *testing* berdasarkan metode 10-*fold cross validation.* Pada 10-*fold cross validation*, data dibagi menjadi 10 *fold* kemudian data dibagi menjadi training testing dengan perbandingan 90:10 dengan metode stratifikasi.

Tabel 4.12 adalah hasil rata-rata dari tingkat akurasi, presisi, *recall*, dan AUC pada data sebelum *oversampling* SMOTE dan setelah dilakukan *oversampling* SMOTE dengan menggunakan metoode 10-*fold cross validation*.

Tabel 4.12 Nilai rata-rata ketepatan klasifikasi dengan metode10-*Cross Fold Validation*

NUMBER OF FEATURE	**KRITERIA PENILAIAN**	**ORIGINAL**	**SMOTE**
ALL FEATURE	Akurasi	0.9010	0.9270
	Presisi	0.8750	0.9530
	Recall	0.8990	0.9290
	AUC	0.5476	0.9300
1500 *FEATURE*	Akurasi	0.9090	0.9390
	Presisi	0.8990	0.9510
	Recall	0.9090	0.9370
	AUC	0.6006	0.9335
500 *FEATURE*	Akurasi	0.9580	0.9560
	Presisi	0.9570	0.9640
	Recall	0.9580	0.9560
	AUC	0.8611	0.9386

Berdasarkan Tabel 4.12 dapat diketahui bahwa nilai rata-rata dari akurasi, presisi, dan *recall* menunjukkan nilai yang hampir sama pada semua *feature selection* dan pada data sebelum dilakukan *oversampling* SMOTE maupun data setelah dilakukan

oversampling SMOTE, namun berbeda dengan nilai AUC yang diperoleh. Evaluasi dari kinerja model klasifikasi dapat juga dihitung dengan menggunakan *Receiver Operating Characteristics* (ROC) dengan mengkombinasikan sensitifitas dan spesifisitas. Area di bawah kurva ROC yang disebut AUC (*Area Under Curve*) dapat digunakan sebagai ukuran kinerja metode klasifikasi. AUC terletak pada interval 0 sampai 1, semakin mendekati 1 maka nilai AUC akan semakin bagus. Nilai AUC pada Tabel 4.8 menunjukkan performansi dari *feature selection* dalam meningkatkan ketepatan klasifikasi. Semakin sedikit jumlah kata yang digunakan maka akan semakin tinggi nilai AUC yang didapatkan yang ditunjukkan dengan warna kuning. Selain itu, performansi dari penggunaan *oversampling* SMOTE juga sangat berpengaruh terhadap meningkatkan ketepatan klasifikasi. Pada Tabel 4.8, ketepatan klasifikasi pada sebelum dan sesudah dilakukan *oversampling* menunjukkan peningkatan yang tinggi, sebagai contoh nilai AUC pada *all feature* sebelum dilakukan *oversampling* adalah 0.5476 sedangkan setelah dilakukan oversampling SMOTE hasilnya meningkat menjadi 93%. Rata-rata ketepatan klasifikasi paling tinggi jika digunakan kata sebanyak 500 dan metode *oversampling* SMOTE dengan nilai akurasi sebesar 0.9560, presisi 0.9640, *recall* 0.9560, dan AUC sebesar 0.9386. Perhitungan pada data SMOTE adalah sebagai berikut.

Berdasarkan Lampiran 11 dapat diketahui bahwa dengan menggunakan kata sebanyak 500 dan metode *oversampling* SMOTE, nilai AUC paling tinggi berada pada *fold* ke-7 dengan hasil sebagai berikut.

Tabel 4.13 *Confusion Matrix* pada *fold*-7 data *Oversampling* SMOTE

Kelas Aktual	**Kelas Prediksi**	
	Negatif	Positif
Negatif	116	6
Positif	0	14

Berdasarkan Tabel 4.13 dapat diketahui bahwa terdapat tweet, sentimen positif yang diprediksikan negatif berjumlah 0

tweet, dan sentimen positif yang diprediksikan positif berjumlah 14 tweet. Berdasarkan hasil *confusion matrix* tersebut didapatkan nilai akurasi sebesar 0.96, presisi sebesar 0.97, *recall* sebesar 0.96 dan AUC sebesar 0.9754.

Selain ditinjau dari data setelah dilakukan *oversampling* SMOTE, dapat ditinjau juga pada data sebelum dilakukan *oversampling* SMOTE (ORIGINAL). Berdasarkan Lampiran 11 nilai AUC paling tinggi jika menggunakan data ORIGINAL dan 500 *feature* juga ditunjukkan oleh *fold* ke – 7 dengan *confusion matrix* sebagai berikut.

Tabel 4.14 *Confusion Matrix* pada *fold*-7 data ORIGINAL

Kelas Aktual	**Kelas Prediksi**	
	Negatif	Positif
Negatif	120	2
Positif	1	13

Berdasarkan Tabel 4.14 dapat diketahui confusion matrix pada data sebelum dilakukan *oversampling* namun digunakan 500 *feature*. Jumlah sentimen negatif yang diklasifikasikan secara benar sentimen negatif adalah sebanyak 120 tweet, sentimen negatif yang salah diklasifikasikan menjadi sentimen positif adalah sebanyak 2 tweet, sentimen positif yang diklasifikasikan salah menjadi sentimen negatif adalah sebanyak 1 tweet, sedangkan jumlah sentimen negatif yang diklasifikasikan secara benar sentimen negatif berjumlah 13 twett. Selanjutnya akan dihitung nilai akurasi, presisi, *recall*, dan AUC pada *confusion matrix* tersebut. Nilai ketepatan ketepatan klasifikasi yang ditinjau dari nilai akurasi adalah sebesar 0.98, presisi 0.98, *recall* 0.98, sedangkan jika ditinjau dengan nilai AUC menghasilkan nilai 0.9561.

Nilai AUC dengan menggunakan 500 *feature* pada data sebelum dilakukan *oversampling* SMOTE dan data setelah dilakukan oversampling SMOTE menghasilkan hasil yang tidak terlalu jauh, sehingga dapat disarankan lebih menggunakan data sebelum digunakan oversampling SMOTE namun tetap

menggunakan 500 *feature*. Pertimbangan yang diberikan adalah untuk mempersingkat waktu pelatihan data.

Model *Naïve Bayes Classifier* (NBC) yang digunakan dalam penelitian ini adalah sebagai berikut.

$$P(X|C_0)P(C_0) = 0.905 \times 0.0006^{f(X_1)} \times 0.0017^{f(X_2)} \times \ldots \times 0.0048^{f(X_{1746})}$$

$$P(X|C_1)P(C_1) = 0.094 \times 0.0417^{f(X_1)} \times 0.0035^{f(X_2)} \times \ldots \times 0.0017^{f(X_{1746})}$$

4.4.2 Klasifikasi Data Tweet Menggunakan *Naïve Bayes Classifier* (NBC) pada Akun @bukabantuan

Sebelum mengklasifikasikan data tweet menggunakan *Naïve Bayes Classifier* pada akun @bukabantuan, dilakukan *feature selection/variabel selection*. *Feature selection* adalah sebuah proses memilih kata (variabel prediktor) yang digunakan untuk pemodelan. Keuntungan menggunakan *feature selection* sebelum melakukan pengolahan data lebih lanjut adalah mengurangi *overfitting*, meningkatkan akurasi, dan mengefisienkan waktu. Berikut adalah langkah untuk memprediksi suatu data *tweet* masuk ke dalam sentimen positif (1) atau negatif (0) pada data tweet @bukabantuan sebelum dilakukan *feature selection*.

Untuk menentukan suatu tweet masuk ke dalam klas negatif atau kelas positif dapat dilihat dari probabilitas $P(X|C_i)P(C_i)$ berikut.

Tabel 4. 15 Probabilitas Klasifikasi NBC pada Bukabantuan

***Testing* Tweet**	**Probabilitas Negatif**	**Probabilitas Positif**	**Keputusan**
1	9.999×10^{-01}	5.412×10^{-08}	Negatif
2	9.926×10^{-01}	7.305×10^{-03}	Negatif
3	9.999×10^{-01}	2.106×10^{-05}	Negatif
4	8.381×10^{-01}	1.618×10^{-01}	Negatif
5	9.911×10^{-01}	8.810×10^{-03}	Negatif
6	9.999×10^{-01}	1.987×10^{-06}	Negatif
7	9.999×10^{-01}	3.764×10^{-08}	Negatif

Tabel 4. 15 Probabilitas Klasifikasi NBC pada Bukabantuan (lanjutan)

Testing **Tweet**	**Probabilitas Negatif**	**Probabilitas Positif**	**Keputusan**
8	2.238×10^{-03}	9.977×10^{-01}	Positif
9	9.998×10^{-01}	1.148×10^{-04}	Negatif
10	6.350×10^{-01}	9.364×10^{-01}	Positif
11	9.959×10^{-01}	4.004×10^{-03}	Negatif
12	9.999×10^{-01}	3.425×10^{-05}	Negatif
13	9.999×10^{-01}	6.046×10^{-05}	Negatif
14	9.999×10^{-01}	1.606×10^{-06}	Negatif
15	9.996×10^{-01}	3.871×10^{-04}	Negatif
⋮	⋮	⋮	
251	9.991×10^{-01}	8.857×10^{-04}	Negatif

Berdasarkan Tabel 4.15 dapat diketahui probabilitas klasifikasi dengan metode Naïve Bayes Classifier (NBC) pada akun @bukabantuan. Jika probabilitas sentimen negatif lebih tinggi jika dibandingkan dengan probabilitas sentimen positif, maka tweet pada data *testing* akan menghasilkan keputusan negatif begitupula sebaliknya. Pada data tweet yang pertama, probabilitas sentimen negatif adalah sebesar 0.999999, sedangkan probabilitas sentimen positif adalah sebesar 0.00000000541. Keputusan yang dapat diambil adalah tweet pertama masuk kedalam sentimen negatif, dikarenakan probabilitas sentimen negatif lebih besar jika dibandingkan dengan probabilitas sentimen positif. Selain itu juga dapat dilihat pada tweet ke-8, probabilitas sentimen negatif adalah sebesar 0.99776103, sedangkan probabilitas sentimen positif adalah sebesar 0.0002238966. sehingga dapat disimpulkan prediksi sentimen pada tweet ke-8 adalah adalah sentimen positif.

Berdasarkan hasil klasifikasi pada Tabel 4.15 dapat diketahui nilai ketepatan klasifikasinya dengan membandingkan Y(prediksi) dan Y(aktual) yang ditunjukkan melalui *confusion matrix* pada Tabel 4.16

Tabel 4.16 *Confusion Matrix* @bukabantuan

Kelas Aktual	Kelas Prediksi	
	Negatif	Positif
Negatif	176	3
Positif	4	68

Tabel 4.16 menunjukkan hasil ketepatan prediksi dengan metode Naïve Bayes Classifier (NBC) pada akun @bukabantuan. Berdasarkan hasil tersebut dapat diketahui bahwa sentimen negatif yang diprediksi secara benar sentimen negatif berjumlah 176 tweet, sentimen negatif yang diprediksi salah menjadi sentimen positif berjumlah 3 tweet, sentimen positif yang diprediksi salah menjadi sentimen negatif berjumlah 4 tweet, dan sentimen negatif yang diprediksi secara benar sentimen negatif berjumlah 68 tweet. Untuk mengetahui berapa ketepatan klasifikasi menggunakan metode *naïve bayes classifier* (NBC) pada akun @bubantuan, maka dilakukan perhitungan dengan melihat akurasi, presisi, *recall*, dan AUC. Nilai akurasi yang didapatkan dari confusion matrix pada Tabel 4.12 adalah sebesar 0.97, presisi sebesar 0.97, *recall* sebesar 0.97 dan AUC sebesar 0.9638.

Berdasarkan hasil akurasi, presisi, *recall*, dan AUC menunjukkan bahwa *Naïve bayes Classifier* (NBC) pada data tweet @bukabantuan sudah cukup baik, sehingga pada kasus ini tidak menggunakan *oversamping* SMOTE (*Synthetic Minority Oversampling TEchnique*). Langkah selanjutnya adalah membandingkan hasil tingkat akurasi dengan membagi data *training* dan *testing* berdasarkan metode 10-*fold cross validation.* Pada 10-*fold cross validation*, data dibagi menjadi 10 *fold* kemudian data dibagi menjadi *training testing* dengan perbandingan 90:10 dengan metode sampling stratifikasi.

Tabel 4.17 adalah hasil rata-rata dari tingkat akurasi, presisi, *recall*, dan AUC pada data sebelum dilakukan *oversampling* SMOTE dengan menggunakan metoode 10-*fold cross validation* yang dicobakan pada semua *feature*, 1500 *feature*, dan 500 *feature*.

Tabel 4.17 Nilai rata-rata ketepatan klasifikasi dengan metode10-*Cross Fold Validation pada Akun @bukabantuan*

NUMBER OF FEATURE	**KRITERIA PENILAIAN**	**KETEPATAN KLASIFIKASI**
ALL FEATURE	Akurasi	0.9380
	Presisi	0.9400
	Recall	0.9380
	AUC	0.9279
1500 *FEATURE*	Akurasi	0.9440
	Presisi	0.9470
	Recall	0.9460
	AUC	0.9344
500 *FEATURE*	Akurasi	0.9600
	Presisi	0.9610
	Recall	0.9610
	AUC	0.9492

Berdasarkan hasil pada Tabel 4.17 dapat diketahui performansi dari *feature selection* dalam *Naïve Bayes Classifier* (NBC) pada akun @bukabantuan. Dengan perhitungan rata-rata nilai akurasi, presisi, dan *recall* pada Tabel 4.13 dapat diketahui bahwa pemilihan *feature selection* sangat berpengaruh dalam meningkatkan ketepatan klasifikasi yang ditunjukkan dengan warna kuning. Semakin sedikit jumlah *feature* yang digunakan, maka rata-rata akurasi, presisi, *recall*, dan AUC juga akan meningkat. Namun pada kasus ini, tidak terlalu banyak perbedaan ketepatan klasifikasi jika menggunakan all *feature*, 1500 *feature*, dan 500 *feature*. Jika dilihat dari berbagai kriteria penilaian ketepatan klasifikasi, maka nilai akurasi, presisi, *recall*, dan AUC paling tinggi jika menggunakan 500 *feature*, dengan nilai akurasi sebesar 0.96, presisi sebesar 0.9610, *recall* sebesar 0.9610, dan AUC sebesar 0.9494. Pengurangan jumlah *feature* dapat mengurangi waktu pelatihan, dan memudahkan proses pengklasifikasian.

Berdasarkan Lampiran 12 dapat diketahui bahwa dengan menggunakan 500 *feature*, nilai AUC paling tinggi berada pada *fold* ke-9 dengan hasil *confusion matrix* sebagai berikut.

Tabel 4.18 *Confusion Matrix* pada *fold*-9 Akun @bukabantuan

Kelas Aktual	**Kelas Prediksi**	
	Negatif	Positif
Negatif	182	2
Positif	4	60

Tabel 4.18 menunjukkan pengelompokan hasil prediksi yang telah dilakukan dengan *Naïve Bayes Classifier* (NBC) pada akun @bukabantuan. Berdasarkan tabel *confusion matrix* tersebut dapat diketahui bahwa sentimen negatif yang diklasifikasikan secara benar sentimen negatif adalah sebesar 183 tweet, sentimen negatif yang diklasifikasikan salah menjadi sentimen positif hanya berjumlah 2 tweet, sentimen positif yang diklasifikasikan salah menjadi sentimen negatif hanya berjumlah 4 tweet, sedangkan sentimen negatif yang diklasifikasikan secara benar sentimen positif berjumlah 60 tweet. Langkah selanjutnya adalah menghitung ketepatan klasifikasi yang dapat ditinjau dari beberapa kriteria yaitu akurasi, presisi, *recall*, dan AUC. Nilai ketepatan klasifikasi yang dilihat dari akurasi adalah sebesar 0.98, presisi sebesar 0.98, *recall* sebesar 0.98 dan nilai AUC sebesar 0.9814. Dari hasil perhitungan tersebut dapat dikatakan bahwa ketepatan klasifikasi dengan menggunakan *Naïve Bayes Classifier* (NBC) pada akun @bukabantuan menunjukkan hasil yang bagus.

Model *Naïve Bayes Classifier* (NBC) yang digunakan dalam penelitian ini adalah sebagai berikut.

$$P(X|C_0)P(C_0) = 0.8944 \times 0.0117^{f(X_1)} \times 0.0036^{f(X_2)} \times \ldots \times 0.0018^{f(X_{1746})}$$

$$P(X|C_1)P(C_1) = 0.1055 \times 0.00667^{f(X_1)} \times 0.0085^{f(X_2)} \times \ldots \times 0.00102^{f(X_{1746})}$$

4.5 Klasifikasi Data Tweet Menggunakan *Artificial Neural Network* (ANN)

Klasifikasi data tweet pada akun @tokopediacare dan @bukabantuan dengan metode Artificial Neural Network menggunakan jaringan *multilayer perceptron*. Metode pelatihan yang digunakan adalah backpropagation dengan 1 *hidden layer*. Backpropagation secara berulang memproses data *training*, membandingkan prediksi jaringan untuk setiap *tuple* dengan nilai target yang diketahui. Jumlah neuron pada *hidden layer* telah ditentukan, yaitu sebanyak 1,2,3,4,5,6,7,8,9,10. *Activation function* yang digunakan adalah *logistic* (sigmoid). Metode yang digunakan untuk mengupdate semua parameter (weight dan bias) menggunakan *Stochastic Gradient Descent* (SGD)

4.5.1 Klasifikasi Data Tweet Menggunakan *Artificial Neural Network* (ANN) pada Akun @tokopediacare

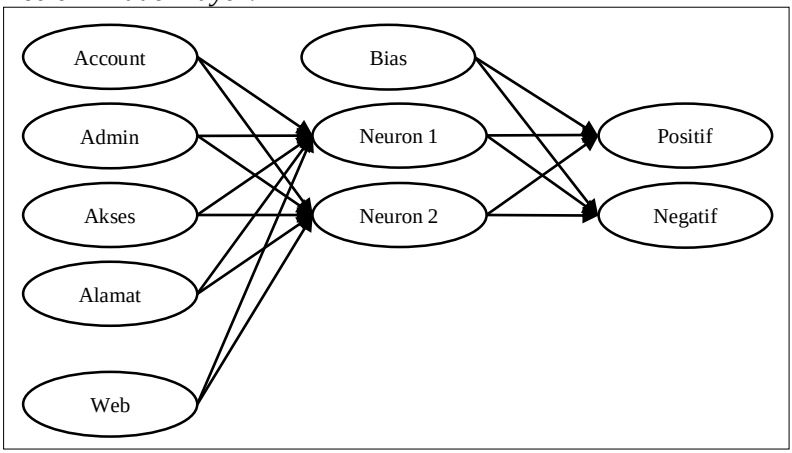
Langkah pertama yang dilakukan untuk klasifikasi data tweet dengan metode *Artificial Neural Network* (ANN) adalah mengetahui jumlah neuron yang menghasilkan ketepatan klasifikasi terbaik dengan membandingkan kriteria ketepatan klasifikasi seperti *akurasi*, *presisi*, *recall*, dan AUC. Selain itu juga dilakukan pemilihan *feature* yang terbaik untuk dilakukan klasifikasi. Hasil dari pemilihan *feature* dan jumlah neuron ditunjukkan pada Tabel 4.19

Tabel 4.19 Ketepatan Klasifikasi ANN pada Akun @tokopediacare

Neuron	ALL *Feature*				1500 *FEATURE*				500 *FEATURE*			
	A	P	R	AUC	A	P	R	AUC	A	P	R	AUC
1	0.96	0.96	0.96	0.9194	0.96	0.96	0.95	0.8929	0.92	0.93	0.92	0.8036
2	0.93	0.94	0.93	0.8393	0.96	0.96	0.96	0.9469	0.96	0.96	0.96	0.9505
3	0.94	0.95	0.94	0.8571	0.95	0.94	0.94	0.9459	0.95	0.95	0.95	0.8750
4	0.96	0.96	0.95	0.8929	0.96	0.97	0.96	0.9107	0.93	0.93	0.92	0.8214
5	0.93	0.93	0.93	0.8393	0.95	0.94	0.94	0.8571	0.96	0.96	0.96	0.8929
6	0.98	0.98	0.98	0.9464	0.97	0.97	0.97	0.9286	0.96	0.96	0.96	0.9061
7	0.97	0.97	0.97	0.9286	0.96	0.96	0.96	0.8929	0.95	0.95	0.95	0.8750
8	0.96	0.96	0.96	0.8929	0.97	0.97	0.97	0.9286	0.95	0.95	0.95	0.8750
9	0.97	0.97	0.97	0.9286	0.97	0.96	0.96	0.9107	0.95	0.95	0.95	0.8750
10	0.97	0.97	0.97	0.9418	0.96	0.97	0.96	0.9107	0.95	0.95	0.95	0.8750

Berdasarkan hasil ketepatan kalasifikasi dengan metode klasifikasi *Artificial Neural Network* (ANN) pada Tabel 4.20 dapat diketahui performansi dari *feature selection* dalam meningkatkan ketepatan klasifikasi yang ditinjau berdasarkan *akurasi*, *presisi*, *recall* dan AUC. Jika menggunakan *all feature* dapat diketahui bahwa jumlah neuron yang paling baik ketepatan klasifikasinya adalah 6 neuron, dengan nilai akurasi 0.98, presisi 0.98, *recall* 0.98, dan AUC sebesar 0.9464. Jika menggunakan 1500 *feature*, ketepatan klasifikasi yang paling baik ditunjukkan jika menggunakan 2 neuron dengan tingkat akurasi sebesar 0.96, presisi sebesar 0.96, *recall* sebesar 0.96, dan nilai AUC sebesar 0.9496. Jika menggunakan 500 *feature*, ketepatan klasifikasi yang paling tinggi ditunjukkan jika menggunakan 2 neuron dengan nilai akurasi sebesar 0.96, presisi 0.96, *recall* 0.96, dan AUC sebesar 0.9505. Dari hasil yang telah diperoleh dapat disimpulkan bahwa penggunaan *feature selection* dapat meningkatkan ketepatan klasifikasi dengan menggunakan metode klasifikasi *Artificial Neural Network* (ANN). Semakin sedikit jumlah *feature* yang digunakan makan akan semakin tinggi tingkat klasifikasi yang dihasilkan. Ketepatan klasifikasi yang paling baik adalah menggunakan 500 *feature* dan 2 neuron dalam *hidden layer*.

Berikut adalah gambaran dari jaringan *Artificial Neural Network* (ANN) dengan menggunakan 500 *feature* dan 2 neuron dalam *hidden layer*.



Gambar 4.5 Jaringan ANN dengan 500 *feature* dan 2 neuron dalam *hidden layer* pada akun @tokopediacare

Langkah selanjutnya adalah membandingkan hasil tingkat akurasi dengan membagi data *training* dan *testing* berdasarkan metode 10-*fold cross validation.* Pada 10-*fold cross validation*, data dibagi menjadi 10 *fold* kemudian data dibagi menjadi *training testing* dengan perbandingan 90:10 dengan metode sampling stratifikasi. Berikut adalah hasil ketepatan klasifikasi dengan menggunakan 10-*fold cross validation,* 500 *feature*, dan 2 neuron dalam *hidden layer* pada akun @tokopediacare

Tabel 4.20 Ketepatan Klasifikasi metode ANN dengan Menggunakan 10-*Fold Cross Validation* pada Akun @tokopediacare

***Fold* ke-**	**KRITERIA PENILAIAN**			
	Akurasi	**Presisi**	***Recall***	**AUC**
1	0.96	0.96	0.96	0.8293
2	0.95	0.95	0.95	0.7959
3	0.91	0.94	0.91	0.9220
4	0.99	0.99	0.99	0.9959
5	0.96	0.96	0.96	0.8000
6	0.90	0.80	0.80	0.7500
7	0.99	0.99	0.99	0.9602
8	0.90	0.80	0.90	0.8100
9	0.97	0.97	0.97	0.9520
10	0.96	0.96	0.96	0.8214
Rata-rata	**0.949**	**0.932**	**0.939**	**0.8636**

Berdasarkan Tabel 4.21 dapat diketahui ketepatan klasifikasi metode ANN dengan menggunakan 10-Fold pada akun @bukabantuan menunjukkan rata-rata tingkat akurasi adalah sebesar 0.949, presisi 0.932, *recall* 0.939, dan rata-rata AUC adalah sebesar 0.8636. Nilai AUC paling tinggi ditunjukkan oleh *fold* ke 4 yang diwarnai dengan warna kuning dengan *confusion matrix* yang ditunjukkan pada Tabel 4.21

Tabel 4.21 *Confusion Matrix* pada *fold* ke-4 Metode ANN pada Akun @tokopedia

Kelas Aktual	Kelas Prediksi	
	Negatif	Positif
Negatif	122	1
Positif	0	15

Tabel 4.21 menunjukkan performansi dari metode klasifikasi pada *fold* ke-4 dengan metode *Artificial Neural Network* (ANN). Dari tabel tersebut dapat diketahui bahwa sentimen negatif yang diklasifikasikan secara benar sentimen negatif berjumlah 122 *tweet*, sentimen negatif yang diklasifikasikan salah menjadi sentimen positif berjumlah 1 tweet, tidak ada sentimen positif yang diklasifikasikan secara salah menjadi sentimen negatif, dan sentimen positif yang diklasifikasikan secara benar sentimen positif berjumlah 15 tweet. Ketepatan klasifikasi yang dihasilkan dari *confusion matrix* tersebut ditinjau dari akurasi sebesar 0.99, presisi sebesar 0.99, *recall* sebesar 0.99 dan AUC sebesar 0.9959. Model yang digunakan adalah sebagai berikut.

Neuron 1 $\hat{Z}_1 = \frac{1}{1+e^{-(-0.505-1.024x_1+0.241x_2-1.062x_3+\ldots+0.504x_{500})}}$

Neuron 2 $\hat{Z}_1 = \frac{1}{1+e^{-(-1.835-0.641x_1+0.778x_2-2.637x_3+\ldots+0.413x_{500})}}$

Model Neural Network

$$\hat{Y}_{NEGATIF} = \frac{1}{1+e^{-(-0.521-1.762x_1+0.450x_2-0.003x_3+\ldots+0.453x_{500})}}$$

$$\hat{Y}_{POSITIF} = \frac{1}{1+e^{-(-2.228-0.027x_1+0.104x_2-0.541x_3+\ldots+0.530x_{500})}}$$

4.4.2 Klasifikasi Data Tweet Menggunakan *Artificial Neural Network* (ANN) pada Akun @bukabantuan

Langkah pertama yang dilakukan untuk klasifikasi data tweet dengan metode *Artificial Neural Network* (ANN) adalah mengetahui jumlah neuron yang menghasilkan ketepatan klasifikasi terbaik dengan membandingkan kriteria ketepatan

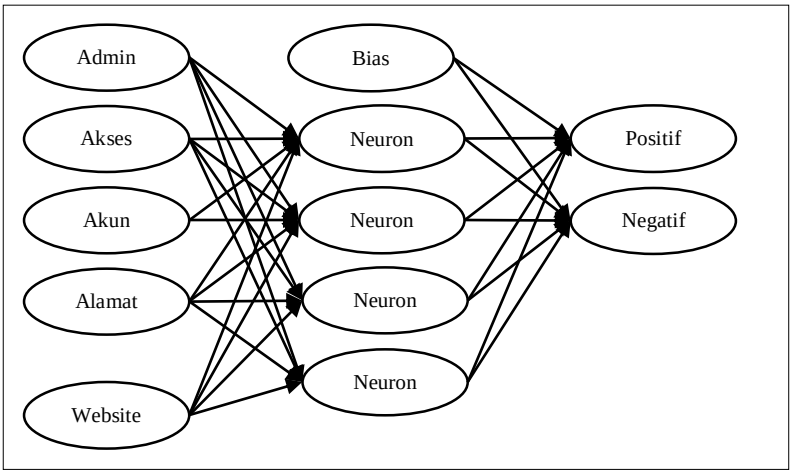
klasifikasi seperti *akurasi, presisi, recall*, dan AUC. Selain itu juga dilakukan pemilihan *feature* yang terbaik untuk dilakukan klasifikasi. Hasil dari pemilihan *feature* dan jumlah neuron ditunjukkan pada Tabel 4.22

Tabel 4.22 Ketepatan Klasifikasi ANN pada Akun @bukabantuan

Neuron	ALL *FEATURE*				1500 *FEATURE*				500 *FEATURE*			
	A	P	R	AUC	A	P	R	AUC	A	P	R	AUC
1	0.98	0.98	0.98	0.9625	0.98	0.98	0.98	0.9708	0.98	0.98	0.98	0.9736
2	0.98	0.98	0.98	0.9749	0.98	0.98	0.98	0.9708	0.98	0.98	0.98	0.9708
3	0.98	0.98	0.98	0.9736	0.98	0.98	0.98	0.9736	0.98	0.98	0.98	0.9694
4	0.98	0.98	0.98	0.9694	0.98	0.98	0.98	0.9694	0.98	0.98	0.98	0.9764
5	0.98	0.98	0.98	0.9764	0.98	0.98	0.98	0.9764	0.98	0.98	0.98	0.9666
6	0.98	0.98	0.98	0.9625	0.98	0.98	0.98	0.9694	0.98	0.98	0.98	0.9694
7	0.98	0.98	0.98	0.9625	0.98	0.98	0.98	0.9625	0.98	0.98	0.98	0.9764
8	0.98	0.98	0.98	0.9694	0.98	0.98	0.98	0.9694	0.98	0.98	0.98	0.9694
9	0.98	0.98	0.98	0.9694	0.98	0.98	0.98	0.9694	0.98	0.98	0.98	0.9625
10	0.98	0.98	0.98	0.9736	0.98	0.98	0.98	0.9625	0.98	0.98	0.98	0.9625

Berdasarkan Tabel 4.23 dapat diketahui nilai ketepatan klasifikasi dengan metode klasifikasi *Artificial Neural Network* (ANN). *Feature selection* dalam kasus ini tidak begitu memberikan pengaruh yang berbeda terhadap ketepatan klasifikasi. Jika dilihat dari kriteria ketepatan klasifikasi akurasi, presisi, dan recall, hasil yang didapat pada semua *feature* menunjukkan hasil yang sama yaitu akurasi 0.98, presisi 0.98, dan recall sebesar 0.98. Namun jika dilihat dari AUC, nilai AUC dengan menggunakan 500 *feature* lebih baik jika dibandingkan dengan menggunakan semua *feature*. Jika menggunakan semua *feature*, nilai AUC yang dihasilkan adalah sebesar 0.9749, sedangkan jika menggunakan 500 *feature*, nilai AUC yang dihasilkan adalah sebesar 0.9764. Berdasarkan hasil ketepatan klasifikasi pada Tabel 4.18 dapat disimpulkan bahwa sebaiknya menggunakan data dengan 500 *feature* dan 4 neuron dikarenakan dapat menyingkat waktu pelatihan.

Berikut adalah gambaran dari jaringan *Artificial Neural Network* (ANN) dengan menggunakan 500 *feature* dan 2 neuron dalam *hidden layer* pada akun @bukabantuan.



Gambar 4.6 Jaringan ANN dengan 500 *feature* dan 4 neuron dalam *hidden layer* pada akun @bukabantuan

Langkah selanjutnya adalah membandingkan hasil tingkat akurasi dengan membagi data *training* dan *testing* berdasarkan metode 10-*fold cross validation.* Pada 10-*fold cross validation*, data dibagi menjadi 10 *fold* kemudian data dibagi menjadi *training testing* dengan perbandingan 90:10 dengan metode sampling stratifikasi. Berikut adalah hasil ketepatan klasifikasi dengan menggunakan 10-*fold cross validation*, 500 *feature*, dan 4 neuron dalam *hidden layer* pada akun @bukabantuan.

Tabel 4.23 Ketepatan Klasifikasi metode ANN dengan Menggunakan 10-*Fold Cross Validation* pada Akun @bukabantuan

Fold ke-	KRITERIA PENILAIAN			
	Akurasi	**Presisi**	**Recall**	**AUC**
1	0.95	0.94	0.94	0.8973
2	0.93	0.93	0.93	0.8715
3	0.98	0.98	0.98	0.9538
4	0.94	0.94	0.94	0.8769
5	0.98	0.98	0.98	0.9715
6	0.98	0.98	0.98	0.9816

Tabel 4.23 Ketepatan Klasifikasi metode ANN dengan Menggunakan 10-*Fold Cross Validation* pada Akun @bukabantuan (lanjutan)

7	0.96	0.96	0.96	0.9204
8	0.95	0.95	0.95	0.9189
9	0.97	0.97	0.97	0.9709
10	0.96	0.96	0.96	0.9294
Rata-rata	0.960	0.959	0.959	0.92922

Berdasarkan hasil pada Tabel 4.24 dapat diketahui ketepatan klasifikasi jika menggunakan metode 10-*fold cross validation.* Rata-rata ketepatan klasifikasi yang ditinjau berdasarkan nilai akurasi adalah sebesar 0.96, presisi 0.959, dan recall sebesar 0.959 dan nilai AUC sebesar 0.92922. Nilai AUC paling tinggi ditunjukkan pada *fold* ke-9 dengan nilai akurasi sebesar 0.97, presisi sebesar 0.97, recall sebesar 0.97, dan nilai AUC sebesar 0.9709. Confusion matrix pada fold ke-9 dapat ditunjukkan pada Tabel 4.24.

Tabel 4.24 *Confusion Matrix* pada *fold* ke-9 Metode ANN pada Akun @bukabantuan

Kelas Aktual	**Kelas Prediksi**	
	Negatif	Positif
Negatif	181	5
Positif	2	62

Berdasarkan Tabel 4.24 dapat diketahui performansi dari metode ANN dalam melakukan prediksi klasifikasi pada data twitter @bukabantuan. Sentimen negatif yang diklasifikasikan secara benar sentimen negatif adalah sebanyak 181 tweet, tweet negatif yang diklasifikasikan secara salah menjadi sentimen positif sebanyak 5, sentimen positif yang diklasifikasikan salah menjadi sentimen negatif sebanyak 2 tweet, sedangkan sentimen positif yang diklasifikasikan secara benar sentimen positif adalah sebanyak 62 tweet. Dari hasil *confusion matrix* pada Tabel 4.20 dapat diketahui tingkat ketepatan klasifikasi yang ditinjau dari nilai

akurasi yaitu sebesar 0.97, presisi sebesar 0.97, recall sebesar 0.97 dan nilai AUC sebesar 0.9709

Neuron 1 $\hat{Z}_1 = \frac{1}{1+e^{-(-0.1667-1.027x_1+0.254x_2+1.124x_3+\ldots+2.024x_{500})}}$

⋮

Neuron 4 $\hat{Z}_1 = \frac{1}{1+e^{-(-2.482+0.239x_1+0.568x_2+0.657x_3+\ldots+0.572x_{500})}}$

Model Neural Network

$$\hat{Y}_{NEGATIF} = \frac{1}{1+e^{-(-2.007+0.001x_1+0.621x_2-1.031x_3+\ldots+0.455x_{500})}}$$

$$\hat{Y}_{POSITIF} = \frac{1}{1+e^{-(-0.375-0.112x_1+0.495_2-0.492x_3+\ldots+0.002x_{500})}}$$

4.5 Perbandingan Performansi Metode Klasifikasi

Perbandingan performansi metode klasifikasi dilakukan agar dapat menentukan metode yang paling baik diantara Naïve Bayes Classifier (NBC) dan *Artificial Neural Network* (ANN) untuk menentukan sentimen konsumen belanja *online* pada akun @tokopediacare dan akun @bukabantuan. Evaluasi dari kinerja metode klasifikasi dihitung dengan menggunakan *Receiver Operating Characteristics* (ROC) yang mengkombinasikan nilai sensitifitas dan spesifisitas. Area di bawah kurva ROC yang disebut AUC (*Area Under Curve*) dapat digunakan sebagai ukuran kinerja metode klasifikasi. AUC terletak pada interval 0 sampai 1, semakin mendekati 1 maka nilai AUC akan semakin bagus. Tabel 4.25 menunjukkan perbandingan performansi metode klasifikasi yang ditinjau berdasarkan Area Under Curve (AUC).

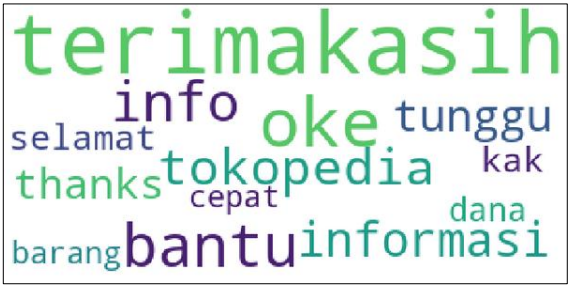
Tabel 4.25 Perbandingan Performansi Metode Klasifikasi dengan AUC

Metode Klasifikasi	Akun	SEMUA KATA	1500 KATA	500 KATA
NBC	TokopediaCare	0.5476	0.6006	0.8611
	BukaBantuan	0.9279	0.9344	0.9492
ANN	TokopediaCare	0.9464	0.9469	0.9505
	BukaBantuan	0.9749	0.9764	0.9764

Berdasarkan Tabel 4.26 dapat diketahui perbandingan performansi dari metode klasifikasi *Naïve Bayes Classifier* (NBC) dan *Artificial Neural Network* (ANN), dapat disimpulkan bahwa dengan pengukuran AUC metode klasifikasi yang paling baik dalam melakukan klasifikasi pada kedua akun belanja *online* adalah *Artificial Neural Network* (ANN) dengan 500 *feature*. Semakin sedikit jumlah *feature* yang digunakan, makan akan semakin tinggi pula ketepatan klasifikasi yang dihasilkan. Jika dilihat berdasarkan akun @TokopediaCare dapat disimpulkan terjadi kenaikan tingkat klasifikasi jika menggunakan metode klasifikasi ANN dibandingkan menggunakan metode NBC.

4.6 Visualisasi *Wordcloud*

Pendapat dari konsumen belanja *online* di Tokopedia dan Bukalapak dapat divisualisasikan dalama bentuk *Wordcloud. Word cloud* merupakan representasi grafis dari sebuah dokumen yang dilakukan dengan *plotting* kata-kata yang sering muncul pada sebuah dokumen pada ruang dua dimensi. Frekuensi dari kata yang sering muncul ditunjukkan melalui ukuran huruf kata tersebut. Semakin besar ukuran kata menunjukkan semakin besar frekuensi kata tersebut mundul dalam dokumen. Visualisasi *wordcloud* dari komentar tweet Tokopedia positif dan negatif ditunjukkan pada Gambar 4.7 dan Gambar 4.8.



Gambar 4.7 *Wordcloud* Sentimen Positif Tokopedia

Berdasarkan Gambar 4.7 dapat diketahui bahwa tiga kata yang sering muncul pada sentimen positif akun Tokopedia adalah terimakasih, bantu, dan informasi. Peningkatan pelayanan tentang bantuan dan memberikan informasi kepada masyarakat dari akun @tokopediacare perlu dipertahankan dan diberi apresiasi. Selain respon positif, respon negatif juga perlu diketahui oleh Tokopedia untuk meningkatkan pelayanan.

Gambar 4.8 *Wordcloud* Sentimen Negatif Tokopedia

Berdasarkan Gambar 4.8 dapat diketahui bahwa sentimen negatif dari konsumen Tokopedia menunjukkan bahwa kata barang adalah kata yang paling banyak dibicarakan. Hal ini dikarenakan Tokopedia banyak menjual barang daripada menjual jasa sehingga barang yang dijual di Tokopedia harus lebih diperhatikan. Visualisasi *wordcloud* dari komentar tweet Bukalapak positif dan negatif ditunjukkan pada Gambar 4.9 dan Gambar 4.10.

Gambar 4.9 *Wordcloud* Sentimen Positif Bukalapak

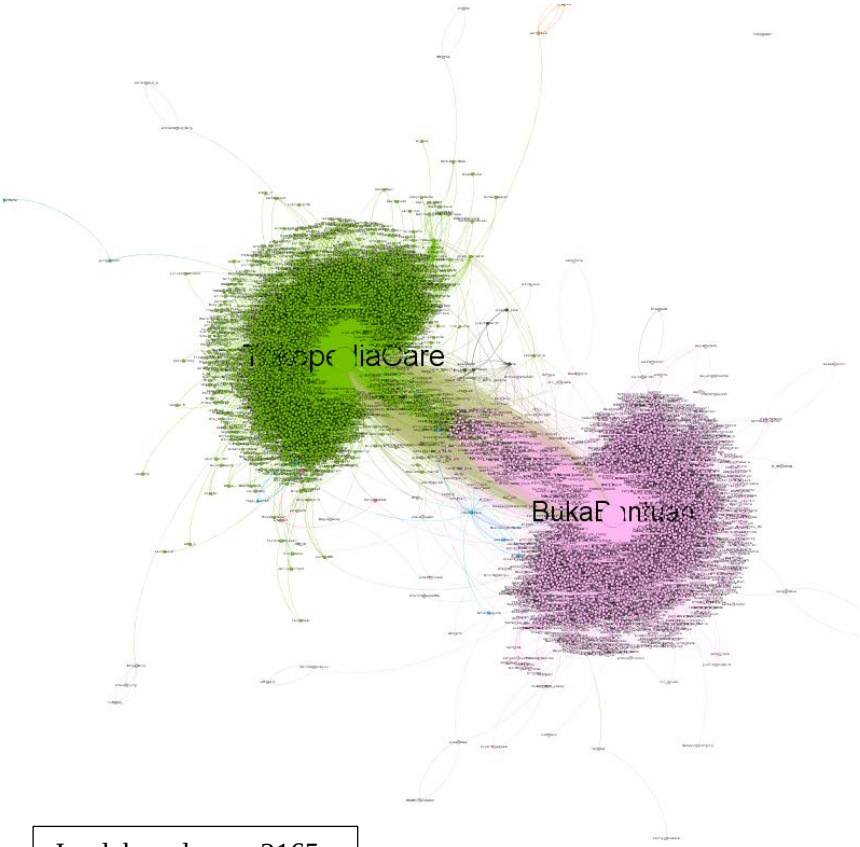
Berdasarkan Gambar 4.9 dapat diketahui bahwa kata yang sering muncul pada sentimen positif akun Bukalapak adalah transaksi dan admin. Peningkatan pelayanan tentang transaksi pembayaran dan pelayanan admin dari akun @bukabantuan perlu dipertahankan dan diberi apresiasi. Selain respon positif, respon negatif juga perlu diketahui oleh Bukalapak untuk meningkatkan pelayanan.

Gambar 4.10 *Wordcloud* Sentimen Negatif Bukalapak

Berdasarkan Gambar 4.10 dapat diketahui bahwa sentimen negatif dari konsumen Bukalapak menunjukkan bahwa kata tolong dan transaksi adalah kata yang paling banyak dibicarakan. Hal ini dikarenakan Bukalapak banyak menjual barang daripada menjual jasa. Selain itu, Gambar 4.10 juga menggambarkan bahwa kata transaksi yang dijual di Tokopedia harus lebih diperhatikan.

4.7 *Social Network Analysis* (SNA)

Informasi yang didapatkan dari twitter tidak dapat menggambarkan struktur komunikasi dan tingkat partisipasi dari setiap pelanggan. Oleh karena itu diperlukan suatu metode yang dapat menilai atau memeriksa pola interaksi pelanggan Tokopedia dan Bukalapak. *Social Network Analysis* (SNA) merupakan salah satu metode untuk menganalisis pola interaksi pelanggan Tokopedia dan Bukalapak yang ditunjukkan pada Gambar 4.11.



Jumlah node	: 2165
Jumlah Edge	: 4542
Communalities	: 26
Graph Density	: 0.01

Gambar 4.11 Social Network Analysis antara Konsumen Tokopedia dan Bukalapak

Social Network Analysis (SNA) dapat memetakan relasi antar orang, organisasi, topik, lokasi, dan intensitas informasi lainnya. Node atau titik di dalam jaringan menggambarkan orang, organisasi, lokasi, atau entitas informasi. Garis sambungan antar titik menggambarkan relasi antar titik.Gambar 4.11 menunjukkan *graph* SNA yang menunjukkan relasi antara konsumen, relasi antara konsumen dana kun @tokopedia dan @bukalapak dan secara visual dapat diketahui bahwa terdapat 2 pusat akun yaitu akun twitter @tokopediacare dan akun twitter @bukabantuan. Dari gambar tersebut juga dapat diketahui pola interaksi antara konsumen tokopedia dan bukalapak pada akun twitter @tokopediacare dan @bukabantuan. Pada gambar tersebut menunjukkan pola interaksi jika terdapat minimal jumlah interaksi yang dilakukan antar akun/orang sebanyak 2 interaksi. Titik-titik hitam adalah akun twitter. Garis berwarna hijau adalah garis yang menggambarkan hubungkan antara konsumen tokopedia terhadap *central* akun @tokopediacare, sedangkan garis berwarna merah muda adalah garis yang menggambarkan hubungan antara konsumen bukalapak terhadap *central* akun @bukabantuan. Pada kasus ini, Jumlah node yang dihasilkan dari graph SNA pada akun twitter @tokopedia dan @bukabantuan adalah 2165 node. Sedangkan jumlah edge/relasi adalah sebesar 4542. *Communalities* yang terbentuk adalah sebanyak 26. Nilai density yang terbentuk dalam graph diatas adalah 0.01. Density adalah nilai kepadatan yang mengacu pada "koneksi" antar akun. Pada kasus ini dapat dilihat bahwa nilai density akun konsumen terhadap central akun cenderung rapat. Ukuran yang digunakan dalam penentuan aktor dalam penelitian ditinjau berdasarkan *degree centrality*, *betweenness centrality*, dan *closeness centrality* yang dijabarkan dalam Tabel 4.26

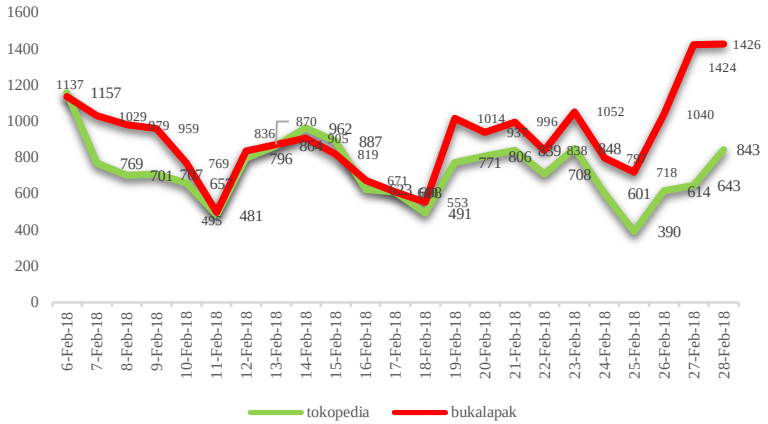
Tabel 4.26 Ukuran Penentuan Aktor pada SNA

Degree Centrality		*Clossenes Centrality*		*Betwenness Centrality*	
Akun	**Nilai**	**Akun**	**Nilai**	**Akun**	**Nilai**
TokopediaCare	4344	BukaBantuan	0.51619	TokopediaCare	12180138.00
BukaBantuan	4267	TokopediaCare	0.51000	BukaBantuan	11981509.08
Tokopedia	74	Krissapto	0.49720	rikkya16com	76398.23
bukalapak	36	YuniNasir_	0.49708	Krissapto	54231.24
Jntexpressid	22	Kalebls	0.49684	YuniNasir_	49826.30
Gojekindonesia	16	201metre	0.49671	Kalebls	41764.45
JNECare	16	ALWANSHOP21	0.49671	Bukalapak	40969.68

Berdasarkan Tabel 4.26 dapat diketahui ukuran penentuan aktor pada *graph Social Network Analysis* (SNA). Ukuran penentuan aktor yang pertama adalah *degree centrality*. Nilai *degree centrality* menunjukkan kedekatan aktor utama yang paling aktif atau memiliki interaksi paling banyak dalam suatu jaringan. Jika ditinjau dari *degree centrality*, TokopediaCare adalah akun yang memiliki lebih banyak interaksi jika dibandingkan dengan akun BukaBantuan, kemudian disusul oleh akun tokopedia, bukalapak, jntexpressid, gojekindonesia, dan JNTCare.

Ukuran penentuan aktor yang kedua adalah *closeness centrality* yang mengukur kedekatan sebuah aktor dengan aktor yang lain. Jika ditinjau dari *closeness centrality*, BukaBantuan adalah aktor yang memiliki kedekatan dengan aktor-aktor lain jika dibandingkan dengan TokopediaCare. Nilai *closeness centrality* dari BukaBantuan adalah sebesar 0.51619.

Ukuran penentuan aktor yang ketiga adalah *betweenness centrality*. *Betweenness centrality* adalah suatu aktor yang menjadi jembatan dalam suatu jaringan atau dengan kata lain adalah aktor yang berpotensi memiliki pengendalian atas interaksi antara dua aktor yang tidak berdekatan dan menjadi titik yang menhubungkan antara dua klaster yang berbeda. Pada Gambar 4.10 dapat diketahui bahwa terdapat dua aktor penting yang menjadi penghubung dari semua jaringan yaitu akun TokopediaCare dan BukaBantuan. Berdasarkan data yang telah diperoleh, pada Gambar 4.12 dapat dijelaskan rincian jumlah interaksi tiap hari antara akun TokopediaCare dan akun BukaBantuan.



Gambar 4.12 Jumlah Interaksi antara Kedua Akun Belanja *Online* dengan Konsumen

Berdasarkan Gambar 4.12 dapat diketahui jumlah interaksi antara kedua akun *costumer care* belanja *online* @tokopediacare dan @bukabantuan. Secara visual dapat diketahui bahwa interaksi @bukabantuan lebih tinggi jika dibandingkan dengan akun belanja *online* @tokopediacare. secara visual dapat diketahui bahwa interaksi antara konsumen dan kedua akun belanja *online* cenderung stabil, namun ada sedikit peningkatan pada akhir bulan. Hal ini dikarenakan bahwa orang cenderung banyak berbelanja pada akhir bulan.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis yang telah diperoleh, kesimpulan yang didapatkan dari penelitian ini adalah sebagai berikut.

1. Konsumen yang memberikan tanggapan, pendapat, kritik, saran dan masalah *complain* lebih banyak ditujukan pada akun @bukabantuan dari pada akun @tokopediacare. Jumlah sentimen data pada akun @tokopediacare dan @bukabantuan bersifat *imbalanced*, sehingga ketepatan klasifikasi perlu dilihat berdasarkan AUC dan dilakukan *oversampling* SMOTE.
2. Performansi dari *feature selection* dapat meningkatkan ketepatan klasifikasi dengan metode Naïve Bayes Classifier (NBC) pada kedua akun belanja *online* @TokopediaCare dan @BukaBantuan. Nilai ketepatan ketepatan klasifikasi dengan menggunakan 500 *feature* yang ditinjau berdasarkan nilai AUC adalah sebesar 0.9561 untuk akun @TokopediaCare, sedangkan nilai AUC pada hasil prediksi klasifikasi data tweet akun @BukaBantuan adalah sebesar 0.9814.
3. Performansi *feature selection* dengan metode Artificial Neural Network (ANN) tidak memberikan kenaikan yang tinggi terhadap tingkat klasifikasi, namun pada penelitian ini tetap digunakan *feature selection* dikarenakan mempertimbangkan waktu pelatihan yang lebih singkat dan efisien. Hasil nilai AUC pada akun @TokopediaCare dengan menggunakan ANN adalah sebesar 0.9959, sedangkan nilai AUC pada akun @BukaBantuan adalah sebesar 0.9709.

4. Berdasarkan visualisasi wordcloud, sentimen positif yang banyak dibicarakan oleh konsumen pada akun @TokopediaCare adalah terimakasih, bantu, dan informasi. Sedangkan sentimen negatif yang paling banyak dibicarakan oleh konsumen adalah barang. Untuk sentimen positif pada akun @BukaBantuan, kata-kata yang banyak dibicarakan oleh masyarakat adalah terimakasih dan admin, sedangkan sentimen negatif yang banyak dibicarakan oleh konsumen adalah tolong, dan transaksi.
5. Berdasarkan hasil graph dari Social Network Analysis dapat disimpulkan bahwa akun @TokopediaCare lebih banyak melakukan interaksi jika dibandingkan dengan akun @BukaBantuan.

5.2 Saran

Berdasarkan hasil analisis yang telah dijelaskan, maka saran yang dapat diberikan pada penelitian yang akan dating adalah sebagai berikut.

1. Pada penelitian klasifikasi teks, membutuhkan data sentimen yang cukup besar untuk mengurangi resiko dari *imbalanced data*. Pengklasifikasian sentimen awal sebaiknya dilakukan dengan banyak orang sehingga dapat mengurangi obyektifitas pada hasil sentimen.
2. Pada proses *steeming*, perlu menambahkan kata kunci khusus untuk Bahasa sehari-hari dikarenakan pada data tweet banyak menggunakan data sehari-hari.

DAFTAR PUSTAKA

Agresti, Alan. 2007. *An Introduction to Categorical Data Analysis*. Second Editon. New Jersey: John Wiley & Sons.

Ary D & Fithriasari K, (2016). *Aplikasi Text Mining untuk Automasi Klasifikasi Artikel dalam Majalah Online Wanita Menggunakan Naïve Bayes Classifier (NBC) dan Artificial Neural Network (ANN)*. Journal of Institut Teknologi Sepuluh Nopember Surabaya.

Aslam, S (2018, January 1). *Twitter by The Number: Stats, Demographics & Fun Facts*. Retrieved from Omnicore: https://www.omnicoreagency.com/twitter-statistics/

Carvalho, G., de Matos, D. M., & Rocio, V. (2007, November). Document Retrieval for Question Answering: A Quantitative Evaluation of Text Preprocessing. In *Proceedings of the ACM first Ph. D. workshop in CIKM* (pp. 125-130). ACM.

Castella, Q., & Sutton, C. (2014). Word Storm: Multiples of Word Clouds for Visual Comparison of Documents.

Dragut, E., Fang, F., Sistla, P., Yu, C., & Meng, W. (2009). Stop Word and Related Problem in Web Interface Integration. *VLDB Endowment*.

Faust, K., & Wasserman, S. (1994). *Social Network Analysis*. New York: Cambridge University Press.

Fieldman, R., & Sanger, J. (2006). *The Text Mining Hand Book*. New York: Cambridge University Press

Fithriasari, K., Iriawan, N., Ulama, B., Kuswanto, H., (2013). *Prediction of hourly rainfall using Bayesian neural network with adjusting procedure*. 3rd Basic Science Internationa; Conference. Surabaya.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning* (Vol. 1). Springer, Berlin: Springer series in statistics.

Gokgoz, E., & Subasi, A. (2015). Comparison of Decision Tree Algorithms for EMG Signal Classification Using DWT. *Biomedical Signal Processing and Control, 18*, 138-144.

Gupta, V., & Lehal, G. S. (2009). A Survey of Text Mining Techniques and Applications. *Journal of emerging technologies in web intelligence, 1*(1),

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques (2nd Edition)*. San Francisco: Morgan Kaufmann Publisher.

Hotho, A., Numberger, A., & Paas, G. (2005). *A Brief Survey of Text Mining*. Kassel: University of Kassel.

Index, T. P. (2017, September 6). *The Connected Consumer*. Retrieved from Tetra Pak Index: https://assets.tetrapak.com/static/documents/about/tetra-pak-index2017.pdf

Indriani, A. (2014). Klasifikasi Data Forum dengan Menggunakan Metode Naïve Bayes Classifier. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*

J. Scott, 1992, Social Network Analysis, Newbury Park CA: Sage.

Kumar, S., Morstatter, F., & Liu, H. (2013). *Twitter Data Analytics*. New York: Springer.

Kusumadewi, S. (2003). Artificial Intelligence (Teknik dan Aplikasinya).*Yogyakarta: Graha Ilmu*.

L. C. Freeman, 1979, "Centrality in social networks: I. conceptual clarification", Social Networks, vol. 1 p.215.

Liu, B. (2010). *Handbook of Natural Language Processing Second Edition*. Boca Raton: CRC Press.

Meinanda, M. H., Annisa, M., Muhandri, N., & Suryadi, K. (2009). Prediksi Masa Studi Sarjana dengan Artificial Neural Network. *Internetworking Indonesia Journal, 1*(2), 31-35.

Moraes, R., Valiati, J. F., & Neto, W. P. G. (2013). Document-Level Sentiment Classification: An Empirical Comparison

Between SVM and ANN. *Expert Systems with Applications*, *40*(2), 621-633.

Novita, Dita (2015). *Analisis Positioning Marketplace Tokopedia, Bukalapak, Qoo10, Rakuten, Elevenia, Lamido Berdasarkan Persepti Konsumen.* Telkom University : Skripsi Manajemen Bisnis Telekomunikasi dan Informatika.

Ollie. (2008). *Membuat toko online dengan Multiply*. Jakarta: Media Kita.

Prilianti, K. R., & Wijaya, H. (2014). Aplikasi Text Mining untuk Automasi Penentuan Tren Topik Skripsi dengan Metode K-Means Clustering. *Jurnal Cybermatika*, *2*(1).

Pujiadi & Widyaiswara. (2013). *Penggunaan Metode Artificial Neural Network dengan Algoritma Sel Organizing Maps untuk Membantu Guru dalam Melakukan Pemetaan Soal Ujian* Nasional. Diakses pada 14 September 2015, dari URL:http://www.lpmpjateng.go.id/web/index.php/arsip/karya-tulis-ilmiah/798-penggunaan-metode-artificial-neural-network-dengan-algoritma-self-organizing-maps-untuk-membantu-gur

Rifqi, N., Maharani, W., & Shaufiah. (2011). Analisis dan Implementasi Klasifikasi Data Mining Menggunakan Jaringan Syaraf Tiruan dan Evolution Strategis. *Konferensi Nasional Sistem dan Informatika.*

Septiana, N., Ridok, A., & Dewi, C. (2013). Pengelompokan Dokumen Berita Berbahasa Indonesia Menggunakan Fuzzy C-Means. *Respository Jurnal Mahasiswa PTIIK UB, 1(7)*

Sun, Y., Kamel, M. S., & Wang, Y. (2006). Boosting for Learning Multiple Classes with Im-balanced Class Distribution. *Sixth International Conference on Data Mining (ICDM'06)*, 421 – 431.

Tan, Ah-Hwee. (1999). Text Mining. *The State of The Art and The Challenges*. Singapore: Kent Ridge Digital Labs.

Tsvetovat, Maksim., Kouznetsov, Alexander. (2011). "Social Network Analysis for Startups". USA:O'reilly Media.

Vapnik, V. (2013). *The Nature of Statistical Learning Theory*. Springer Science & Business Media.

W. Nooy, A. Mrvar, and V. Batagelj, 2005, Exploratory Social Network Analysis with Pajek, Cambridge University Press.

Weiss, S. M. (2010). *Text Mining: Predictive Methods for Analyzing Unstructural Information*. New York: Springer.

Wiki Book. (2011). *Social Network Analysis: Theory and Applications.* Tersedia: https://www.politaktiv.org/documents/10157/29141/SocNet_TheoryApp.pdf [08 Februari 2018].

Yang, Y. & Pedersen, J. O (1997). *A comparative Study on Feature Selection in Text Categorization.* ICML '97 Proceedings of the Fourteenth International Conference on Machine Learning. San Francisco : USA.

Zaki, M. J., & Meira, W. (2014). *Data Mining And Analysis: Fondations and Algorithms.* New York: Cambridge University Press.

LAMPIRAN

Lampiran 1. Syntax Crawling Data

```
library(twitteR)
#ID Twitter API
consumer_key <-'NSoTBX1avojA6CIHcHGjVBDjV'
consumer_secret<-'eqAUlMFxPxJlYlGk7WNxOxLRB4awGV1wcn10492HIgnliLiXP3'
access_token <-'90558142-T9NjU3bgidYQiGVjY10zsWmhH9IxxChyrrM7S3ILd'
access_secret <-'btkXiA0sDvU4JIb0YnkoTrbN9DfCyteobyqI4mXHribb4'

#Login Twitter API
setup_twitter_oauth(consumer_key, consumer_secret, access_token,
access_secret)

#mencari hasil
tokopedia<-
searchTwitter("tokopediacare",n=150000,lang="id",since="2018-3-
1",until="2018-3-1")
write.csv(twListToDF(tokopedia), file="datatokopedia1.csv")
```

Lampiran 2. Hasil Crawling Data Twitter

Nomer	Text	Favorite Count	Created	…	Screen Name
1	@TokopediaCare Sepertinya ada kendala trx, sy melakukan 2x trx kartu kredit dan 1 via tt bank. Apakah yg kartu kred… https://t.co/FYDYIWHYEd	0	2/6/2018 23:59		Mintari_shining
2	@ady_sut --SHIPMENT FORWARDED TO DESTINATION [TANGERANG, HUB NEGLASARI]". Untuk itu, kami sarankan kamu menunggu hi… https://t.co/7QTroVxm4a	0	2/6/2018 23:49		TokopediaCare
3	@ady_sut Hi Ady, mohon maaf sebelumnya. Berdasarkan pengecekan kami untuk transaksi kamu dengan invoice INV/2018021… https://t.co/OHkj8QDxSZ	0	2/6/2018 23:48		TokopediaCare
4	@lincung Hi Adhim. Baik, apabila nantinya kamu memiliki kendala atau pertanyaan, jangan ragu untuk menghubungi kami… https://t.co/zTWvCnO4oB	0	2/6/2018 23:33		TokopediaCare
5	@dedihartonoo -- pengecekan pesanan kamu secara langsung dengan cara mengakses web Sicepat dan menginput nomor resi… https://t.co/jLubD5xjEr	0	2/6/2018 23:30		TokopediaCare
6	@dedihartonoo Hi Dedi, mohon maaf atas ketidaknyamanannya. Kami mengerti kekhawatiranmu, namun jika kami cek untuk… https://t.co/EBbd7kxu8p	0	2/6/2018 23:30		TokopediaCare
7	@TokopediaCare terimaksih	0	2/6/2018 23:28		lincung

8	Dear @TokopediaCare sy punya kendala utk nomor order INV/20180212/XVIII/II/135379405 sudah 3 hari semenjak sy order… https://t.co/ypfbjj8zbC	0	2/6/2018 22:59	dedihartonoo
9	@itsjustfahira_ @itsjustfahira_ pertanyaan lainnya silakan hubungi kami kembali. Terima kasih ^MZA	0	2/6/2018 22:56	TokopediaCare
10	@itsjustfahira_ @itsjustfahira_ Hai Fahira, dengan senang hati kami bisa membantu. Terima kasih telah mempercayakan… https://t.co/bOtxbGD9Yt	0	2/6/2018 22:56	TokopediaCare
11	@lincung Hi Adhim, mohon maaf sebelumnya. Kamu dapat masuk ke menu Pembelian setelah itu pilih Daftar Transaksi. Si… https://t.co/q7pDbKLEOB	0	2/6/2018 22:56	TokopediaCare
12	@TokopediaCare Baik kalau begitu. Teeima kasih banyak atas infonya :) aangat membantu <ed><U+00A0><U+00BD><ed><U+00B1><U+008D><ed><U+00A0><U+00BD><ed><U+00B1><U+008D>	0	2/6/2018 22:41	itsjustfahira_
13	@TokopediaCare kalo mau liat invoice dari transaksi yang sudah selesai dimana? kok nggak ada saya cari2	0	2/6/2018 22:28	lincung
14	@seha_olshop @seha_olshop -- untuk menghubungi kami kembali ya. Terima kasih ^IRZ	0	2/6/2018 22:23	TokopediaCare
15	@seha_olshop @seha_olshop Hai Toppers. Terima kasih atas kepercayaan kamu terhadap Tokopedia. Kami akan terus berus… https://t.co/s6ayblL68q	0	2/6/2018 22:23	TokopediaCare
16	@mynismar @mynismar -- kunjungi pada link berikut (https://t.co/T4Z8OQrvvX). Apabila kamu mengalami kendala lain di… https://t.co/fokpC5p4uV	0	2/6/2018 22:21	TokopediaCare
17	@mynismar @mynismar -- Tokopedia, sebelum melakukan pembayaran, akan terdapat menu Gunakan Kode Promo/Kupon, silaka… https://t.co/3P5KyNjtn7	0	2/6/2018 22:18	TokopediaCare
18	@mynismar @mynismar -- (https://t.co/0Hug6gaUqa). Kamu juga dapat menggunakan Kupon gratis ongkos kirim yang dimana… https://t.co/xw8EKuLAmN	0	2/6/2018 22:11	TokopediaCare
19	@mynismar @mynismar Hai Nisa, mohon maaf atas ketidaknyamananya. Perlu diketahui masing-masing E-Commerce memiliki… https://t.co/p3jWaUS7at	0	2/6/2018 22:11	TokopediaCare
20	@TokopediaCare min ko belanja di toped gak gratis ongkir kaya situs lain? Caranya gmna biar gratis ongkir?	0	2/6/2018 22:10	mynismar
…	…	…	…	
…	…	…	…	
n	@yoko_the83 @yoko_the83 -- (https://t.co/PJwGKXq3Ts). Terima kasih ^IRZ	0	2/28/2018 00:01	TokopediaCare

Lampiran 3. Preprocessing Data

```
import pandas as pd
import pandas as dataframe
import string
import nltk
import matplotlib as mpl
import matplotlib.pyplot as plt
import re
import sys
import os
import numpy
import seaborn as sns
from nltk.tokenize import word_tokenize
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from IPython.display import display
from wordcloud import WordCloud, STOPWORDS
from pylab import figure, axes, pie, title, show
%matplotlib inline

data = pd.read_csv('tokopediafix.csv', engine='python')
data.Sentimen.value_counts()

pre_text = data['Teks']

datanolink=[]
for line in pre_text:
    result=re.sub(r"http\S+","",line)
    datanolink.append(result)
print (datanolink)

datanort = []
for line in datanolink:
    result = re.sub(r"RT","",line)
    datanort.append(result)
print (datanort)

datanousername = []
for line in datanort:
    result=re.sub(r"@\S+","",line)
    datanousername.append(result)

print (datanousername)
len (datanousername)

data_lower=[]
for line in datanousername:
    a = line.lower()
    data_lower.append(a)

print (data_lower)

factory=StemmerFactory()
stemmer=factory.create_stemmer()
datastemmed=map(lambda x: stemmer.stem(x), data_lower)
databersih=map(lambda     x:     x.translate(str.maketrans('',     '',
string.punctuation)), datastemmed)
```

```
databersih=list(databersih)
print (databersih)

stopwords=open('stopword.txt', 'r').read()

satudata=[]
datafinal=[]
df=[]
for line in databersih:
    wt_data = word_tokenize(line)
    wt_data = [word for word in wt_data if not word in stopwords and
not word[0].isdigit()]
    datafinal.append(wt_data)
    df.append(" ".join(wt_data))
for l in datafinal:
    satudata+= l
final={v: satudata.count(v) for v in set(satudata)}

import csv

with open ('final.csv', 'w', newline="") as csv_file:
    writer= csv.writer(csv_file)
    for key, value in final.items():
        writer.writerow([key, value])

import matplotlib
import matplotlib.pyplot as plt

kata = pd.DataFrame.from_dict(final,orient='index')
hasilsort = kata.sort_values(by=[0])
hasilsortnew = hasilsort.tail(15)

ax = hasilsortnew.plot(kind='bar', figsize=(6,3), color='#98FB98',
title='Frekuensi Kata Terbanyak @tokopediacare', legend=False)
ax.set_ylim(0, 200)

sns.despine(bottom=False, left=False)
```

Lampiran 4. Hasil Preprocessing

```
[['kendala', 'kartu', 'kredit', 'via', 'bank', 'yg', 'kartu'],
['transaksi', 'batal', 'refund', 'toko', 'cash', 'gabisa'], ['bayar',
'blm', 'status', 'pesan'], ['mohon', 'respon', 'kak'], ['tiket',
'malang', 'semarang', 'email', 'transaksi', 'hasil', 'gimana'],
['gagal', 'transaksi', 'baca', 'ongkir', 'info'], ['gmn', 'no', 'resi',
'palsu', 'valid', 'barang', 'kirim', 'tp', 'resi', 'palsu'], ['bayar',
'tiket', 'kereta', 'terang', 'bayar', 'kadaluarsa'], ['beli', 'tiket',
'kereta', 'pake', 'debit', 'visa', 'mandiri', 'trus', 'ganti',
'clickpay', 'muncul', 'bayar', 'gagal'], ['error', 'pilih', 'nominal',
'mtix'], ['order', 'blm', 'pdhal', 'pakai', 'jne', 'yes', 'inv',
'xviii'], ['coba', 'safari', 'browser', 'iphone', 'gogle', 'crome',
'android', 'tdak', 'bisaa'], ['pake', 'aplikasi', 'mozilla', 'chrome',
'explorer', 'browser', 'tetep', 'musti', 'gimana'], ['link','klik',
'duhhh'], ['iya', 'baca', 'indomaret', 'error', 'sistem']]
```

Lampiran 5. *Count Vectorizer dan TFIDF*

```
#Count Vectorizer
from pandas import DataFrame
count_vectorizer = CountVectorizer(min_df=0., max_df=1.0)
X = count_vectorizer.fit_transform(df)
test_data=DataFrame(X.A,
columns=count_vectorizer.get_feature_names())

#Count Vectorizer
from pandas import DataFrame
count_vectorizer = CountVectorizer(min_df=0., max_df=1.0)
X = count_vectorizer.fit_transform(df)
test_data=DataFrame(X.A,
columns=count_vectorizer.get_feature_names())

#TFIDF
from pandas import DataFrame
from sklearn.feature_extraction.text import TfidfTransformer

tfidf = TfidfTransformer(use_idf=True).fit_transform(test_data)
tfidf_baru = (tfidf.toarray())
print (tfidf_baru)
print (tfidf_baru.shape)

[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
(1369, 1746)
```

Lampiran 6. *Feature Selection*

```
#Feature Selection
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

X, Y_feature = tfidf_baru, Y
x_new = SelectKBest(chi2, k=1500).fit_transform(test_data, Y_feature)

#print (x_new.shape)
#print (x_new)
aaa = chi2(X, Y_feature)
print (aaa)
```

Lampiran 7. Hasil *Chi-Square*

Nomer	Nilai Chisquare	P-Value	Critical Value	Keputusan Berdasarkan Pvalue	Keputusan Berdasarkan ChiSquare Tabel
1	0.110068568	0.740066	0.1	Accept H0	Accept H0
2	0.052990027	0.81794	3.841458821	Accept H0	Accept H0
3	0.083140818	0.773085		Accept H0	Accept H0
4	0.146818313	0.701595	P-value	Accept H0	Accept H0
5	0.365495135	0.54547	ChiSquare Tabel	Accept H0	Accept H0
6	0.10545867	0.745375		Accept H0	Accept H0
7	0.059105579	0.807915		Accept H0	Accept H0
8	0.046865277	0.828611		Accept H0	Accept H0
9	0.101486507	0.750053		Accept H0	Accept H0
10	0.043201053	0.835347		Accept H0	Accept H0
11	0.492077291	0.483002		Accept H0	Accept H0
12	0.187154658	0.665295		Accept H0	Accept H0
13	0.069777177	0.791661		Accept H0	Accept H0
14	0.037758103	0.84593		Accept H0	Accept H0
15	0.086007575	0.769315		Accept H0	Accept H0
16	0.064269298	0.799871		Accept H0	Accept H0
17	2.260966271	0.132671		Accept H0	Accept H0
18	0.057729747	0.810121		Accept H0	Accept H0
19	0.061079066	0.804798		Accept H0	Accept H0
20	0.069357447	0.792275		Accept H0	Accept H0
21	0.117485085	0.731778		Accept H0	Accept H0
22	0.069152865	0.792574		Accept H0	Accept H0
23	0.058658711	0.808629		Accept H0	Accept H0
24	0.044113976	0.833642		Accept H0	Accept H0
25	0.044243708	0.833401		Accept H0	Accept H0
26	0.404530527	0.524759		Accept H0	Accept H0
27	0.387576029	0.533576		Accept H0	Accept H0
28	0.194754028	0.658989		Accept H0	Accept H0
29	0.047531962	0.827415		Accept H0	Accept H0
30	0.203616523	0.651817		Accept H0	Accept H0
31	0.134061037	0.714258		Accept H0	Accept H0
32	0.330116259	0.565591		Accept H0	Accept H0
33	0.255898868	0.612952		Accept H0	Accept H0
34	0.388968121	0.532842		Accept H0	Accept H0
35	0.040631129	0.840252		Accept H0	Accept H0
36	0.046596784	0.829095		Accept H0	Accept H0
37	0.155995792	0.69287		Accept H0	Accept H0
…	…	…		…	…
…	…	…		…	…
1746	0.093632066	0.75961		Accept H0	Accept H0

Lampiran 8. *Spliting Data* dan SMOTE

```
#Spliting Data
from sklearn.model_selection import train_test_split,cross_val_score

# Random training and testing data
X_train,X_test,Y_train,Y_test=train_test_split(test_data,Y,test_size=
0.1,shuffle=False)

#Balancing Data
import imblearn
import numpy
import numpy as np
from imblearn.over_sampling import SMOTE

sm = SMOTE()

X_train_res, y_train_res = sm.fit_sample(X_train, Y_train)

unique, counts = np.unique(y_train_res, return_counts=True)
print(list(zip(unique, counts)))
```

Lampiran 9. *Naïve Bayes Classifier*

```
from sklearn.naive_bayes import BernoulliNB
nb = BernoulliNB()

y_score = nb.fit(X_train, Y_train)
y_pred = nb.predict(X_test)
print(confusion_matrix(Y_test,y_pred))
print("Akurasi Score :
{:.2f}".format(accuracy_score(Y_test,y_pred)))
print(classification_report(Y_test,y_pred))

#ROC CURVE AND AUC
import numpy as np
import matplotlib.pyplot as plt
from itertools import cycle
from sklearn.metrics import roc_curve, auc
from scipy import interp

from sklearn import metrics
import pandas as pd
from ggplot import *

#y_pred = nb.predict(X_test)
fpr, tpr, _ = metrics.roc_curve(Y_test, y_pred)

df = pd.DataFrame(dict(fpr=fpr, tpr=tpr))
ggplot(df, aes(x='fpr', y='tpr')) +\
 geom_line() +\
```

```
 geom_abline(linetype='dashed')
auc = metrics.auc(fpr,tpr)
print("Area Under Curve ROC = {:.2f}% ".format(auc * 100))
nb.predict_proba(X_test)
nb.score(X_train, Y_train, sample_weight=None)
each_class = nb.predict_log_proba(X_train)
```

Lampiran 10. *Naïve Bayes Classifier* dengan KFold

```
YB = DataFrame.as_matrix(Y)
print (YB)
from sklearn.model_selection import StratifiedKFold
from sklearn import metrics
X_baru, Y = x_new, YB
kf = StratifiedKFold(n_splits=10, shuffle=False)
kf.get_n_splits(X_baru)
kf.get_n_splits(Y)
cl = BernoulliNB()
sm = SMOTE()
for train, test in kf.split(X_baru, Y):
    (X_baru[train],X_baru[test])
    (Y[train],Y[test])
    a = cl.fit(X_baru[train], Y[train])
    b = cl.predict(X_baru[test])
    fpr, tpr, _ = metrics.roc_curve(Y[test], b)
    auc = metrics.auc(fpr,tpr)
    print("Klasifikasi Naive Bayes pada Data Original")
    print (confusion_matrix(Y[test],b))
    print("Akurasi Score : {:.2f}".format(accuracy_score(Y[test],b)))
    print (classification_report(Y[test],b))
    print("Area Under Curve ROC = {:.2f} ".format(auc * 100))
    print ("-----------------------------------------------------")
    print("Klasifikasi Naive Bayes dengan Oversampling SMOTE")
    X_train_res, y_train_res = sm.fit_sample(X_baru[train], Y[train])
    aa = cl.fit(X_train_res, y_train_res)
    bb = cl.predict(X_baru[test])
    fpr, tpr, _ = metrics.roc_curve(Y[test], bb)
    auc = metrics.auc(fpr,tpr)
    cm = confusion_matrix(Y[test],bb)
    print (cm)
    print("Akurasi Score :
{:.2f}".format(accuracy_score(Y[test],bb)))
    cr = classification_report(Y[test],bb)
    print (cr)
    print("Area Under Curve ROC dengan Oversampling SMOTE = {:.2f}%
".format(auc * 100))

    print ("*************************************************")
```

Lampiran 11. Hasil N*aïve Bayes Classifier* dengan K-Fold Tokopedia

FOLD	DATA	SEMUA KATA				1500 KATA				500 KATA			
		ACC	P	R	AUC	ACC	P	R	AUC	ACC	P	R	AUC
1	ORIGINAL	0.88	0.79	0.88	0.4919	0.88	0.79	0.88	0.4959	0.95	0.95	0.95	0.7959
	SMOTE	0.96	0.96	0.96	0.9171	0.96	0.96	0.96	0.9171	0.97	0.97	0.97	0.9252
2	ORIGINAL	0.88	0.79	0.88	0.4959	0.9	0.91	0.9	0.5355	0.95	0.95	0.95	0.7959
	SMOTE	0.88	0.93	0.88	0.8724	0.89	0.94	0.89	0.9098	0.92	0.95	0.92	0.926
3	ORIGINAL	0.88	0.84	0.88	0.5252	0.88	0.85	0.88	0.5797	0.96	0.96	0.96	0.9463
	SMOTE	0.91	0.94	0.91	0.8927	0.94	0.91	0.92	0.8927	0.94	0.95	0.94	0.9089
4	ORIGINAL	0.91	0.9	0.91	0.6293	0.92	0.91	0.92	0.6626	0.96	0.96	0.96	0.8919
	SMOTE	0.91	0.95	0.92	0.9553	0.92	0.95	0.92	0.9553	0.95	0.97	0.95	0.9715
5	ORIGINAL	0.91	0.89	0.91	0.5959	0.91	0.9	0.91	0.6292	0.96	0.95	0.96	0.8292
	SMOTE	0.94	0.96	0.94	0.938	0.96	0.96	0.96	0.9462	0.96	0.96	0.96	0.9462
6	ORIGINAL	0.9	0.91	0.91	0.5357	0.91	0.92	0.91	0.5714	0.93	0.93	0.93	0.8326
	SMOTE	0.95	0.97	0.95	0.9713	0.96	0.97	0.96	0.9754	0.98	0.98	0.98	0.9561
7	ORIGINAL	0.89	0.84	0.89	0.5275	0.92	0.93	0.92	0.6071	0.98	0.98	0.98	0.9561
	SMOTE	0.91	0.95	0.91	0.9508	0.92	0.95	0.92	0.9549	0.96	0.97	0.96	0.9754
8	ORIGINAL	0.91	0.95	0.91	0.5316	0.92	0.91	0.92	0.6388	0.96	0.96	0.96	0.853
	SMOTE	0.94	0.96	0.94	0.9356	0.96	0.96	0.96	0.9438	0.96	0.96	0.96	0.9163
9	ORIGINAL	0.94	0.92	0.91	0.5714	0.93	0.94	0.93	0.6786	0.96	0.96	0.96	0.8214
	SMOTE	0.92	0.95	0.93	0.9274	0.93	0.95	0.93	0.8999	0.96	0.96	0.96	0.9122
10	ORIGINAL	0.91	0.92	0.91	0.5714	0.92	0.93	0.92	0.6071	0.97	0.97	0.97	0.8888
	SMOTE	0.95	0.96	0.95	0.9397	0.95	0.96	0.95	0.9397	0.96	0.97	0.96	0.9479
Rata-rata	OGIRINAL	0.901	0.875	0.899	0.54758	0.909	0.899	0.909	0.60059	0.958	0.957	0.958	0.86111
	SMOTE	0.927	0.953	0.929	0.93003	0.939	0.951	0.937	0.93348	0.956	0.964	0.956	0.93857

Lampiran 12. Hasil N*aïve Bayes Classifier* dengan K-Fold BukaBantuan

	SEMUA KATA				1500 KATA				500 KATA			
FOLD Ke-	ACC	P	R	AUC	ACC	P	R	AUC	ACC	P	R	AUC
1	0.91	0.91	0.91	0.8781	0.92	0.93	0.92	0.9089	0.95	0.95	0.95	0.93
2	0.93	0.93	0.93	0.9093	0.96	0.96	0.96	0.9354	0.96	0.96	0.96	0.9304
3	0.95	0.95	0.95	0.9327	0.95	0.95	0.95	0.9327	0.96	0.96	0.96	0.9431
4	0.92	0.92	0.92	0.8989	0.92	0.92	0.92	0.8989	0.94	0.94	0.94	0.9223
5	0.94	0.94	0.94	0.937	0.94	0.94	0.94	0.937	0.96	0.96	0.96	0.9554
6	0.91	0.93	0.91	0.9359	0.92	0.94	0.94	0.9412	0.95	0.96	0.96	0.9677
7	0.93	0.93	0.93	0.9216	0.94	0.94	0.94	0.927	0.96	0.96	0.96	0.9531
8	0.94	0.94	0.94	0.9238	0.94	0.94	0.94	0.9211	0.96	0.96	0.96	0.9448
9	0.98	0.98	0.98	0.9814	0.98	0.98	0.98	0.9814	0.98	0.98	0.98	0.9814
10	0.97	0.97	0.97	0.9606	0.97	0.97	0.97	0.9606	0.98	0.98	0.98	0.9633
RATA-RATA	0.938	0.94	0.938	0.92793	0.944	0.947	0.946	0.93442	0.96	0.961	0.961	0.94915

Lampiran 13. *Artificial Neural Network*

```
from sklearn.neural_network import MLPClassifier
nn = MLPClassifier(hidden_layer_sizes=(2,))

y_score = nn.fit(X_train, Y_train)
y_pred = nn.predict(X_test)
print(confusion_matrix(Y_test,y_pred))
print("Akurasi Score : {:.2f}".format(accuracy_score(Y_test,y_pred)))
print(classification_report(Y_test,y_pred))

#ROC CURVE AND AUC
import numpy as np
import matplotlib.pyplot as plt
from itertools import cycle
from sklearn.metrics import roc_curve, auc
from scipy import interp

from sklearn import metrics
import pandas as pd
from ggplot import *

#y_pred = nb.predict(X_test)
fpr, tpr, _ = metrics.roc_curve(Y_test, y_pred)

df = pd.DataFrame(dict(fpr=fpr, tpr=tpr))
ggplot(df, aes(x='fpr', y='tpr')) +\
 geom_line() +\
 geom_abline(linetype='dashed')

auc = metrics.auc(fpr,tpr)

print("Area Under Curve ROC = {:.2f}% ".format(auc * 100))

loss_function = cl.loss_curve_
print (loss_function)

loss_function_pd = pd.DataFrame.from_dict(loss_function)
loss_function_pd.columns = ['loss']
ax = loss_function_pd.plot(kind='line',)

ylim = ([0.69, 0.71])

ax.set_ylim(ylim)
plt.tight_layout()
plt.show()
```

Lampiran 14. *Artificial Neural Network* dengan KFOLD

```
from sklearn.model_selection import StratifiedKFold
from sklearn.neural_network import MLPClassifier
from sklearn import metrics
X_baru, Y = x_new, YB
kf = StratifiedKFold(n_splits=10, shuffle=False)
kf.get_n_splits(X_baru)
kf.get_n_splits(Y)

cl = MLPClassifier(hidden_layer_sizes=(2,))

for train, test in kf.split(X_baru, Y):
    (X_baru[train],X_baru[test])
    (Y[train],Y[test])
    a = cl.fit(X_baru[train], Y[train])
    b = cl.predict(X_baru[test])
    fpr, tpr, _ = metrics.roc_curve(Y[test], b)
    auc = metrics.auc(fpr,tpr)
    print("Klasifikasi Artificial Neural Network dengan 500 Feature")
    print (confusion_matrix(Y[test],b))
    print("Akurasi Score : {:.2f}".format(accuracy_score(Y[test],b)))
    print (classification_report(Y[test],b))
    print("Area Under Curve ROC = {:.2f} ".format(auc * 100))
    print ("---------------------------------------------------------")
```

Lampiran 15. Hasil *Artificial Neural Network* dengan KFOLD pada akun TokopediaCare

Fold ke-	KRITERIA PENILAIAN			
	Akurasi	Presisi	Recall	AUC
1	0.96	0.96	0.96	82.93
2	0.95	0.95	0.95	79.59
3	0.91	0.94	0.91	92.2
4	0.99	0.99	0.99	99.59
5	0.96	0.96	0.96	80
6	0.9	0.8	0.8	75
7	0.99	0.99	0.99	96.02
8	0.9	0.8	0.9	81
9	0.97	0.97	0.97	95.2
10	0.96	0.96	0.96	82.14
Rata-rata	0.949	0.932	0.939	86.367

Lampiran 16. Hasil *Artificial Neural Network* dengan KFOLD pada akun BukaBantuan

Fold ke-	KRITERIA PENILAIAN			
	Akurasi	Presisi	Recall	AUC
1	0.95	0.94	0.94	0.8973
2	0.93	0.93	0.93	0.8715
3	0.98	0.98	0.98	0.9538
4	0.94	0.94	0.94	0.8769
5	0.98	0.98	0.98	0.9715
6	0.98	0.98	0.98	0.9816
7	0.96	0.96	0.96	0.9204
8	0.95	0.95	0.95	0.9189
9	0.97	0.97	0.97	0.9709
10	0.96	0.96	0.96	0.9294
Rata-rata	0.96	0.959	0.959	0.92922

Lampiran 17. Confusion Matrix

Lampiran 17A. Confusion Matrix Tokopedia ALL *FEATURE* dengan Metode Naïve Bayes

```
Klasifikasi Naive Bayes pada Data Original
[[121   2]
 [ 15   0]]
Akurasi Score : 0.88
             precision    recall  f1-score   support

          0       0.89      0.98      0.93       123
          1       0.00      0.00      0.00        15

avg / total       0.79      0.88      0.83       138

Area Under Curve ROC = 49.19
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[119   4]
 [  2  13]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.98      0.97      0.98       123
          1       0.76      0.87      0.81        15

avg / total       0.96      0.96      0.96       138

Area Under Curve ROC dengan Oversampling SMOTE = 91.71%
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   1]
 [ 15   0]]
Akurasi Score : 0.88
```

```
             precision    recall  f1-score   support

          0       0.89      0.99      0.94       123
          1       0.00      0.00      0.00        15

avg / total       0.79      0.88      0.84       138

Area Under Curve ROC = 49.59
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[108  15]
 [  2  13]]
Akurasi Score : 0.88
             precision    recall  f1-score   support

          0       0.98      0.88      0.93       123
          1       0.46      0.87      0.60        15

avg / total       0.93      0.88      0.89       138

Area Under Curve ROC dengan Oversampling SMOTE = 87.24%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[121   2]
 [ 14   1]]
Akurasi Score : 0.88
             precision    recall  f1-score   support

          0       0.90      0.98      0.94       123
          1       0.33      0.07      0.11        15

avg / total       0.84      0.88      0.85       138
Area Under Curve ROC = 52.52
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[113  10]
 [  2  13]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       0.98      0.92      0.95       123
          1       0.57      0.87      0.68        15

avg / total       0.94      0.91      0.92       138

Area Under Curve ROC dengan Oversampling SMOTE = 89.27%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   1]
 [ 11   4]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       0.92      0.99      0.95       123
          1       0.80      0.27      0.40        15

avg / total       0.90      0.91      0.89       138
```

```
Area Under Curve ROC = 62.93
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[112  11]
 [  0  15]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       1.00      0.91      0.95       123
          1       0.58      1.00      0.73        15

avg / total       0.95      0.92      0.93       138

Area Under Curve ROC dengan Oversampling SMOTE = 95.53%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[121   1]
 [ 12   3]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       0.91      0.99      0.95       122
          1       0.75      0.20      0.32        15

avg / total       0.89      0.91      0.88       137

Area Under Curve ROC = 59.59
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[115   7]
 [  1  14]]
Akurasi Score : 0.94
             precision    recall  f1-score   support

          0       0.99      0.94      0.97       122
          1       0.67      0.93      0.78        15

avg / total       0.96      0.94      0.95       137

Area Under Curve ROC dengan Oversampling SMOTE = 93.80%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   0]
 [ 13   1]]
Akurasi Score : 0.90
             precision    recall  f1-score   support

          0       0.90      1.00      0.95       122
          1       1.00      0.07      0.13        14

avg / total       0.91      0.90      0.87       136

Area Under Curve ROC = 53.57
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[115   7]
```

```
 [  0  14]]
Akurasi Score : 0.95
             precision    recall  f1-score   support

          0       1.00      0.94      0.97       122
          1       0.67      1.00      0.80        14

avg / total       0.97      0.95      0.95       136

Area Under Curve ROC dengan Oversampling SMOTE = 97.13%
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[120   2]
 [ 13   1]]
Akurasi Score : 0.89
             precision    recall  f1-score   support

          0       0.90      0.98      0.94       122
          1       0.33      0.07      0.12        14

avg / total       0.84      0.89      0.86       136

Area Under Curve ROC = 52.75
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[110  12]
 [  0  14]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       1.00      0.90      0.95       122
          1       0.54      1.00      0.70        14

avg / total       0.95      0.91      0.92       136

Area Under Curve ROC dengan Oversampling SMOTE = 95.08%
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[121   1]
 [ 13   1]]
Akurasi Score : 0.90
             precision    recall  f1-score   support

          0       0.90      0.99      0.95       122
          1       0.50      0.07      0.12        14

avg / total       0.86      0.90      0.86       136
Area Under Curve ROC = 53.16
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[115   7]
 [  1  13]]
Akurasi Score : 0.94
             precision    recall  f1-score   support

          0       0.99      0.94      0.97       122
          1       0.65      0.93      0.76        14
```

```
avg / total       0.96      0.94      0.95       136
Area Under Curve ROC dengan Oversampling SMOTE = 93.56%
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   0]
 [ 12   2]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       0.91      1.00      0.95       122
          1       1.00      0.14      0.25        14

avg / total       0.92      0.91      0.88       136

Area Under Curve ROC = 57.14
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[113   9]
 [  1  13]]
Akurasi Score : 0.93
             precision    recall  f1-score   support

          0       0.99      0.93      0.96       122
          1       0.59      0.93      0.72        14

avg / total       0.95      0.93      0.93       136
Area Under Curve ROC dengan Oversampling SMOTE = 92.74%
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   0]
 [ 12   2]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       0.91      1.00      0.95       122
          1       1.00      0.14      0.25        14

avg / total       0.92      0.91      0.88       136

Area Under Curve ROC = 57.14
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[116   6]
 [  1  13]]
Akurasi Score : 0.95
             precision    recall  f1-score   support

          0       0.99      0.95      0.97       122
          1       0.68      0.93      0.79        14
avg / total       0.96      0.95      0.95       136

Area Under Curve ROC dengan Oversampling SMOTE = 93.97%
**************************************************
```

Lampiran 17B. *Confusion Matrix* Tokopedia 1500 *FEATURE* dengan Metode Naïve Bayes

```
Klasifikasi Naive Bayes pada Data Original
[[122   1]
 [ 15   0]]
Akurasi Score : 0.88
             precision    recall  f1-score   support

          0       0.89      0.99      0.94       123
          1       0.00      0.00      0.00        15

avg / total       0.79      0.88      0.84       138

Area Under Curve ROC = 49.59
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[119   4]
 [  2  13]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.98      0.97      0.98       123
          1       0.76      0.87      0.81        15

avg / total       0.96      0.96      0.96       138

Area Under Curve ROC dengan Oversampling SMOTE = 91.71%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[123   0]
 [ 14   1]]
Akurasi Score : 0.90
             precision    recall  f1-score   support

          0       0.90      1.00      0.95       123
          1       1.00      0.07      0.12        15

avg / total       0.91      0.90      0.86       138

Area Under Curve ROC = 53.33
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[109  14]
 [  1  14]]
Akurasi Score : 0.89
             precision    recall  f1-score   support

          0       0.99      0.89      0.94       123
          1       0.50      0.93      0.65        15

avg / total       0.94      0.89      0.90       138

Area Under Curve ROC dengan Oversampling SMOTE = 90.98%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
```

```
[[118   5]
 [ 12   3]]
Akurasi Score : 0.88
             precision    recall  f1-score   support

          0       0.91      0.96      0.93       123
          1       0.38      0.20      0.26        15

avg / total       0.85      0.88      0.86       138

Area Under Curve ROC = 57.97
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[113  10]
 [  2  13]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       0.98      0.92      0.95       123
          1       0.57      0.87      0.68        15

avg / total       0.94      0.91      0.92       138

Area Under Curve ROC dengan Oversampling SMOTE = 89.27%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   1]
 [ 10   5]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       0.92      0.99      0.96       123
          1       0.83      0.33      0.48        15

avg / total       0.91      0.92      0.90       138

Area Under Curve ROC = 66.26
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[112  11]
 [  0  15]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       1.00      0.91      0.95       123
          1       0.58      1.00      0.73        15

avg / total       0.95      0.92      0.93       138

Area Under Curve ROC dengan Oversampling SMOTE = 95.53%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[121   1]
 [ 11   4]]
Akurasi Score : 0.91
             precision    recall  f1-score   support
```

```
          0       0.92      0.99      0.95       122
          1       0.80      0.27      0.40        15

avg / total       0.90      0.91      0.89       137

Area Under Curve ROC = 62.92
------------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[117   5]
 [  1  14]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.99      0.96      0.97       122
          1       0.74      0.93      0.82        15

avg / total       0.96      0.96      0.96       137

Area Under Curve ROC dengan Oversampling SMOTE = 94.62%
******************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   0]
 [ 12   2]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       0.91      1.00      0.95       122
          1       1.00      0.14      0.25        14

avg / total       0.92      0.91      0.88       136

Area Under Curve ROC = 57.14
------------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[116   6]
 [  0  14]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       1.00      0.95      0.97       122
          1       0.70      1.00      0.82        14

avg / total       0.97      0.96      0.96       136

Area Under Curve ROC dengan Oversampling SMOTE = 97.54%
******************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   0]
 [ 11   3]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       0.92      1.00      0.96       122
          1       1.00      0.21      0.35        14

avg / total       0.93      0.92      0.89       136
```

```
Area Under Curve ROC = 60.71
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[111  11]
 [  0  14]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       1.00      0.91      0.95       122
          1       0.56      1.00      0.72        14

avg / total       0.95      0.92      0.93       136

Area Under Curve ROC dengan Oversampling SMOTE = 95.49%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[121   1]
 [ 10   4]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       0.92      0.99      0.96       122
          1       0.80      0.29      0.42        14

avg / total       0.91      0.92      0.90       136

Area Under Curve ROC = 63.88
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[117   5]
 [  1  13]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.99      0.96      0.97       122
          1       0.72      0.93      0.81        14

avg / total       0.96      0.96      0.96       136

Area Under Curve ROC dengan Oversampling SMOTE = 94.38%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   0]
 [  9   5]]
Akurasi Score : 0.93
             precision    recall  f1-score   support

          0       0.93      1.00      0.96       122
          1       1.00      0.36      0.53        14

avg / total       0.94      0.93      0.92       136

Area Under Curve ROC = 67.86
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[115   7]
 [  2  12]]
```

```
Akurasi Score : 0.93
             precision    recall  f1-score   support

          0       0.98      0.94      0.96       122
          1       0.63      0.86      0.73        14

avg / total       0.95      0.93      0.94       136

Area Under Curve ROC dengan Oversampling SMOTE = 89.99%
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   0]
 [ 11   3]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       0.92      1.00      0.96       122
          1       1.00      0.21      0.35        14

avg / total       0.93      0.92      0.89       136

Area Under Curve ROC = 60.71
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[116   6]
 [  1  13]]
Akurasi Score : 0.95
             precision    recall  f1-score   support

          0       0.99      0.95      0.97       122
          1       0.68      0.93      0.79        14

avg / total       0.96      0.95      0.95       136

Area Under Curve ROC dengan Oversampling SMOTE = 93.97%
**************************************************
```

Lampiran 17C. *Confusion Matrix* Tokopedia 500 *FEATURE* dengan Metode Naïve Bayes

```
[[122   1]
 [  6   9]]
Akurasi Score : 0.95
             precision    recall  f1-score   support

          0       0.95      0.99      0.97       123
          1       0.90      0.60      0.72        15

avg / total       0.95      0.95      0.94       138

Area Under Curve ROC = 79.59
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[121   2]
```

```
 [  2  13]]
Akurasi Score : 0.97
             precision    recall  f1-score   support

          0       0.98      0.98      0.98       123
          1       0.87      0.87      0.87        15

avg / total       0.97      0.97      0.97       138
Area Under Curve ROC dengan Oversampling SMOTE = 92.52%
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   1]
 [  6   9]]
Akurasi Score : 0.95
             precision    recall  f1-score   support
          0       0.95      0.99      0.97       123
          1       0.90      0.60      0.72        15
avg / total       0.95      0.95      0.94       138

Area Under Curve ROC = 79.59
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[113  10]
 [  1  14]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       0.99      0.92      0.95       123
          1       0.58      0.93      0.72        15

avg / total       0.95      0.92      0.93       138
Area Under Curve ROC dengan Oversampling SMOTE = 92.60%
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[118   5]
 [  1  14]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.99      0.96      0.98       123
          1       0.74      0.93      0.82        15

avg / total       0.96      0.96      0.96       138

Area Under Curve ROC = 94.63
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[117   6]
 [  2  13]]
Akurasi Score : 0.94
             precision    recall  f1-score   support
          0       0.98      0.95      0.97       123
          1       0.68      0.87      0.76        15

avg / total       0.95      0.94      0.94       138

Area Under Curve ROC dengan Oversampling SMOTE = 90.89%
```

```
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[121   2]
 [  3  12]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.98      0.98      0.98       123
          1       0.86      0.80      0.83        15

avg / total       0.96      0.96      0.96       138

Area Under Curve ROC = 89.19
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[116   7]
 [  0  15]]
Akurasi Score : 0.95
             precision    recall  f1-score   support

          0       1.00      0.94      0.97       123
          1       0.68      1.00      0.81        15

avg / total       0.97      0.95      0.95       138

Area Under Curve ROC dengan Oversampling SMOTE = 97.15%
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[121   1]
 [  5  10]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.96      0.99      0.98       122
          1       0.91      0.67      0.77        15

avg / total       0.95      0.96      0.95       137

Area Under Curve ROC = 82.92
--------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[117   5]
 [  1  14]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.99      0.96      0.97       122
          1       0.74      0.93      0.82        15

avg / total       0.96      0.96      0.96       137

Area Under Curve ROC dengan Oversampling SMOTE = 94.62%
**************************************************
Klasifikasi Naive Bayes pada Data Original
[[116   6]
 [  4  10]]
Akurasi Score : 0.93
```

```
             precision    recall  f1-score   support

          0       0.97      0.95      0.96       122
          1       0.62      0.71      0.67        14
avg / total       0.93      0.93      0.93       136

Area Under Curve ROC = 83.26
------------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[120   2]
 [  1  13]]
Akurasi Score : 0.98
             precision    recall  f1-score   support

          0       0.99      0.98      0.99       122
          1       0.87      0.93      0.90        14

avg / total       0.98      0.98      0.98       136
Area Under Curve ROC dengan Oversampling SMOTE = 95.61%
******************************************************
Klasifikasi Naive Bayes pada Data Original
[[120   2]
 [  1  13]]
Akurasi Score : 0.98
             precision    recall  f1-score   support

          0       0.99      0.98      0.99       122
          1       0.87      0.93      0.90        14

avg / total       0.98      0.98      0.98       136
Area Under Curve ROC = 95.61
------------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[116   6]
 [  0  14]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       1.00      0.95      0.97       122
          1       0.70      1.00      0.82        14

avg / total       0.97      0.96      0.96       136

Area Under Curve ROC dengan Oversampling SMOTE = 97.54%
******************************************************
Klasifikasi Naive Bayes pada Data Original
[[121   1]
 [  4  10]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.97      0.99      0.98       122
          1       0.91      0.71      0.80        14

avg / total       0.96      0.96      0.96       136

Area Under Curve ROC = 85.30
```

```
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[119   3]
 [  2  12]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.98      0.98      0.98       122
          1       0.80      0.86      0.83        14

avg / total       0.96      0.96      0.96       136

Area Under Curve ROC dengan Oversampling SMOTE = 91.63%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[122   0]
 [  5   9]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.96      1.00      0.98       122
          1       1.00      0.64      0.78        14

avg / total       0.96      0.96      0.96       136

Area Under Curve ROC = 82.14
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[118   4]
 [  2  12]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.98      0.97      0.98       122
          1       0.75      0.86      0.80        14

avg / total       0.96      0.96      0.96       136

Area Under Curve ROC dengan Oversampling SMOTE = 91.22%
*****************************************************
Klasifikasi Naive Bayes pada Data Original
[[121   1]
 [  3  11]]
Akurasi Score : 0.97
             precision    recall  f1-score   support

          0       0.98      0.99      0.98       122
          1       0.92      0.79      0.85        14

avg / total       0.97      0.97      0.97       136

Area Under Curve ROC = 88.88
-----------------------------------------------------
Klasifikasi Naive Bayes dengan Oversampling SMOTE
[[118   4]
 [  1  13]]
Akurasi Score : 0.96
```

```
             precision    recall  f1-score   support

          0       0.99      0.97      0.98       122
          1       0.76      0.93      0.84        14

avg / total       0.97      0.96      0.96       136

Area Under Curve ROC dengan Oversampling SMOTE = 94.79%
**************************************************
```

Lampiran 17D. *Confusion Matrix* BukaBantuan ALL *FEATURE* dengan Metode Naïve Bayes

```
Klasifikasi Naive Bayes pada Data Original
[[175  11]
 [ 12  53]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       0.94      0.94      0.94       186
          1       0.83      0.82      0.82        65

avg / total       0.91      0.91      0.91       251

Area Under Curve ROC = 87.81
------------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[178   8]
 [  9  56]]
Akurasi Score : 0.93
             precision    recall  f1-score   support

          0       0.95      0.96      0.95       186
          1       0.88      0.86      0.87        65

avg / total       0.93      0.93      0.93       251

Area Under Curve ROC = 90.93
------------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[181   5]
 [  7  58]]
Akurasi Score : 0.95
             precision    recall  f1-score   support

          0       0.96      0.97      0.97       186
          1       0.92      0.89      0.91        65

avg / total       0.95      0.95      0.95       251

Area Under Curve ROC = 93.27
------------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[177   9]
 [ 10  55]]
Akurasi Score : 0.92
```

```
             precision    recall  f1-score   support

          0       0.95      0.95      0.95       186
          1       0.86      0.85      0.85        65

avg / total       0.92      0.92      0.92       251

Area Under Curve ROC = 89.89
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[174  12]
 [  4  61]]
Akurasi Score : 0.94
             precision    recall  f1-score   support

          0       0.98      0.94      0.96       186
          1       0.84      0.94      0.88        65

avg / total       0.94      0.94      0.94       251

Area Under Curve ROC = 93.70
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[165  21]
 [  1  64]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       0.99      0.89      0.94       186
          1       0.75      0.98      0.85        65

avg / total       0.93      0.91      0.92       251

Area Under Curve ROC = 93.59
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[174  12]
 [  6  59]]
Akurasi Score : 0.93
             precision    recall  f1-score   support

          0       0.97      0.94      0.95       186
          1       0.83      0.91      0.87        65

avg / total       0.93      0.93      0.93       251

Area Under Curve ROC = 92.16
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[178   8]
 [  7  57]]
Akurasi Score : 0.94
             precision    recall  f1-score   support

          0       0.96      0.96      0.96       186
          1       0.88      0.89      0.88        64
```

```
avg / total       0.94      0.94      0.94       250

Area Under Curve ROC = 92.38
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[182   4]
 [  1  63]]
Akurasi Score : 0.98
             precision    recall  f1-score   support

          0       0.99      0.98      0.99       186
          1       0.94      0.98      0.96        64

avg / total       0.98      0.98      0.98       250

Area Under Curve ROC = 98.14
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[182   3]
 [  4  60]]
Akurasi Score : 0.97
             precision    recall  f1-score   support

          0       0.98      0.98      0.98       185
          1       0.95      0.94      0.94        64

avg / total       0.97      0.97      0.97       249

Area Under Curve ROC = 96.06
-----------------------------------------------------
```

Lampiran 17E. *Confusion Matrix* BukaBantuan 1500 *FEATURE* dengan Metode Naïve Bayes

```
Klasifikasi Naive Bayes pada Data Original
[[175  11]
 [  8  57]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       0.96      0.94      0.95       186
          1       0.84      0.88      0.86        65

avg / total       0.93      0.92      0.92       251

Area Under Curve ROC = 90.89
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[182   4]
 [  7  58]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.96      0.98      0.97       186
          1       0.94      0.89      0.91        65
```

```
avg / total       0.96      0.96      0.96       251

Area Under Curve ROC = 93.54
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[181   5]
 [  7  58]]
Akurasi Score : 0.95
             precision    recall  f1-score   support

          0       0.96      0.97      0.97       186
          1       0.92      0.89      0.91        65

avg / total       0.95      0.95      0.95       251
Area Under Curve ROC = 93.27
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[177   9]
 [ 10  55]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       0.95      0.95      0.95       186
          1       0.86      0.85      0.85        65
avg / total       0.92      0.92      0.92       251

Area Under Curve ROC = 89.89
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[174  12]
 [  4  61]]
Akurasi Score : 0.94
             precision    recall  f1-score   support
          0       0.98      0.94      0.96       186
          1       0.84      0.94      0.88        65

avg / total       0.94      0.94      0.94       251

Area Under Curve ROC = 93.70
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[167  19]
 [  1  64]]
Akurasi Score : 0.92
             precision    recall  f1-score   support

          0       0.99      0.90      0.94       186
          1       0.77      0.98      0.86        65

avg / total       0.94      0.92      0.92       251
Area Under Curve ROC = 94.12
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[176  10]
 [  6  59]]
Akurasi Score : 0.94
```

```
             precision    recall  f1-score   support

          0       0.97      0.95      0.96       186
          1       0.86      0.91      0.88        65

avg / total       0.94      0.94      0.94       251
Area Under Curve ROC = 92.70
------------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[177   9]
 [  7  57]]
Akurasi Score : 0.94
             precision    recall  f1-score   support
          0       0.96      0.95      0.96       186
          1       0.86      0.89      0.88        64

avg / total       0.94      0.94      0.94       250

Area Under Curve ROC = 92.11
------------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[182   4]
 [  1  63]]
Akurasi Score : 0.98
             precision    recall  f1-score   support

          0       0.99      0.98      0.99       186
          1       0.94      0.98      0.96        64

avg / total       0.98      0.98      0.98       250

Area Under Curve ROC = 98.14
------------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[182   3]
 [  4  60]]
Akurasi Score : 0.97
             precision    recall  f1-score   support

          0       0.98      0.98      0.98       185
          1       0.95      0.94      0.94        64

avg / total       0.97      0.97      0.97       249

Area Under Curve ROC = 96.06
------------------------------------------------------
```

Lampiran 17F. *Confusion Matrix* BukaBantuan 500 *FEATURE* dengan Metode Naïve Bayes

```
Klasifikasi Naive Bayes pada Data Original
[[180   6]
 [  7  58]]
```

```
Akurasi Score : 0.95
             precision    recall  f1-score   support

          0       0.96      0.97      0.97       186
          1       0.91      0.89      0.90        65

avg / total       0.95      0.95      0.95       251

Area Under Curve ROC = 93.00
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[183   3]
 [  8  57]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.96      0.98      0.97       186
          1       0.95      0.88      0.91        65

avg / total       0.96      0.96      0.96       251

Area Under Curve ROC = 93.04
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[182   4]
 [  6  59]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.97      0.98      0.97       186
          1       0.94      0.91      0.92        65

avg / total       0.96      0.96      0.96       251

Area Under Curve ROC = 94.31
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[180   6]
 [  8  57]]
Akurasi Score : 0.94
             precision    recall  f1-score   support

          0       0.96      0.97      0.96       186
          1       0.90      0.88      0.89        65

avg / total       0.94      0.94      0.94       251

Area Under Curve ROC = 92.23
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[178   8]
 [  3  62]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.98      0.96      0.97       186
          1       0.89      0.95      0.92        65
```

```
avg / total       0.96      0.96      0.96       251

Area Under Curve ROC = 95.54
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[174  12]
 [  0  65]]
Akurasi Score : 0.95
             precision    recall  f1-score   support

          0       1.00      0.94      0.97       186
          1       0.84      1.00      0.92        65

avg / total       0.96      0.95      0.95       251

Area Under Curve ROC = 96.77
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[180   6]
 [  4  61]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.98      0.97      0.97       186
          1       0.91      0.94      0.92        65

avg / total       0.96      0.96      0.96       251

Area Under Curve ROC = 95.31
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[180   6]
 [  5  59]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.97      0.97      0.97       186
          1       0.91      0.92      0.91        64

avg / total       0.96      0.96      0.96       250

Area Under Curve ROC = 94.48
-----------------------------------------------------
Klasifikasi Naive Bayes pada Data Original
[[182   4]
 [  1  63]]
Akurasi Score : 0.98
             precision    recall  f1-score   support

          0       0.99      0.98      0.99       186
          1       0.94      0.98      0.96        64

avg / total       0.98      0.98      0.98       250

Area Under Curve ROC = 98.14
-----------------------------------------------------
```

```
Klasifikasi Naive Bayes pada Data Original
[[183   2]
 [  4  60]]
Akurasi Score : 0.98
             precision    recall  f1-score   support

          0       0.98      0.99      0.98       185
          1       0.97      0.94      0.95        64

avg / total       0.98      0.98      0.98       249

Area Under Curve ROC = 96.33
-----------------------------------------------------
```

Lampiran 18. *Artificial Neural Network* dengan KFOLD 500 *FEATURE*

```
Klasifikasi Artificial Neural Network dengan 500 Feature
[[122   1]
 [  5  10]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.96      0.99      0.98       123
          1       0.91      0.67      0.77        15

avg / total       0.96      0.96      0.95       138

Area Under Curve ROC = 82.93
-----------------------------------------------------
Klasifikasi Artificial Neural Network dengan 500 Feature
[[122   1]
 [  6   9]]
Akurasi Score : 0.95
             precision    recall  f1-score   support

          0       0.95      0.99      0.97       123
          1       0.90      0.60      0.72        15

avg / total       0.95      0.95      0.94       138

Area Under Curve ROC = 79.59
-----------------------------------------------------
Klasifikasi Artificial Neural Network dengan 500 Feature
[[112  11]
 [  1  14]]
Akurasi Score : 0.91
             precision    recall  f1-score   support

          0       0.99      0.91      0.95       123
          1       0.56      0.93      0.70        15

avg / total       0.94      0.91      0.92       138
```

```
Area Under Curve ROC = 92.20
-----------------------------------------------------
Klasifikasi Artificial Neural Network dengan 500 Feature
[[122   1]
 [  0  15]]
Akurasi Score : 0.99
             precision    recall  f1-score   support

          0       1.00      0.99      1.00       123
          1       0.94      1.00      0.97        15

avg / total       0.99      0.99      0.99       138

Area Under Curve ROC = 99.59
-----------------------------------------------------
Klasifikasi Artificial Neural Network dengan 500 Feature
[[122   0]
 [  6   9]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.95      1.00      0.98       122
          1       1.00      0.60      0.75        15

avg / total       0.96      0.96      0.95       137

Area Under Curve ROC = 80.00
-----------------------------------------------------
Klasifikasi Artificial Neural Network dengan 500 Feature
[[122   0]
 [ 14   0]]
Akurasi Score : 0.90
             precision    recall  f1-score   support

          0       0.90      1.00      0.95       122
          1       0.00      0.00      0.00        14

avg / total       0.80      0.90      0.85       136

Area Under Curve ROC = 50.00
-----------------------------------------------------
Klasifikasi Artificial Neural Network dengan 500 Feature
[[121   1]
 [  1  13]]
Akurasi Score : 0.99
             precision    recall  f1-score   support

          0       0.99      0.99      0.99       122
          1       0.93      0.93      0.93        14

avg / total       0.99      0.99      0.99       136

Area Under Curve ROC = 96.02
-----------------------------------------------------
Klasifikasi Artificial Neural Network dengan 500 Feature
[[122   0]
```

```
 [ 14   0]]
Akurasi Score : 0.90
             precision    recall  f1-score   support

          0       0.90      1.00      0.95       122
          1       0.00      0.00      0.00        14

avg / total       0.80      0.90      0.85       136

Area Under Curve ROC = 50.00
-----------------------------------------------------
Klasifikasi Artificial Neural Network dengan 500 Feature
[[119   3]
 [  1  13]]
Akurasi Score : 0.97
             precision    recall  f1-score   support

          0       0.99      0.98      0.98       122
          1       0.81      0.93      0.87        14

avg / total       0.97      0.97      0.97       136
Area Under Curve ROC = 95.20
-----------------------------------------------------
Klasifikasi Artificial Neural Network dengan 500 Feature
[[122   0]
 [  5   9]]
Akurasi Score : 0.96
             precision    recall  f1-score   support

          0       0.96      1.00      0.98       122
          1       1.00      0.64      0.78        14
avg / total       0.96      0.96      0.96       136

Area Under Curve ROC = 82.14
```

Lampiran 19. Surat Pernyataan Pengambilan Data

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika FMKSD ITS:

Nama : Fransiska Kristin Damayanti

NRP : 062116 4500 0035

menyatakan bahwa data yang digunakan dalam Tugas Akhir / ~~Thesis~~ ini merupakan data sekunder yang diambil dari ~~Penelitian / Buku / Tugas Akhir / Thesis /~~ Publikasi lainnya yaitu:

Sumber : Twitter API (*Application Program Interface*)

Keterangan : Data *tweet* dengan *keywords* "Tokopediacare" dan "Bukabantuan"

Surat Pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Mengetahui
Pembimbing Tugas Akhir

Dr. Dra. Kartika Fithriasari, M.Si
NIP. 19691212 199303 2 002

Surabaya, Juli 2018

Fransiska Kristin Damayanti
NRP. 062116 4500 0035

*(coret yang tidak perlu)

(Halaman ini Sengaja Dikosongkan)

BIODATA PENULIS

Penulis bernama lengkap Fransiska Kristin Damayanti merupakan anak keempat dari pasangan Bonifasius Sarni dan Sisilia Sri Harmijati. Penulis lahir di Madiun, pada tanggal 3 Mei 1995. Pendidikan formal yang ditempuh penulis adalah SDK Santo Bavo Madiun, SMPK Santo Yusuf Madiun, SMA N 5 Madiun. Setelah lulus SMA penulis mengikuti seleksi penerimaan mahasiswa baru dan diterima di program studi Diploma III Departemen Statistika Institut Teknologi Sepuluh Nopember pada tahun 2013. Penulis memilih untuk melanjutkan studi guna menempuh gelar sarjana di Departemen Statistika ITS pada tahun 2016. Selama masa perkuliahan penulis aktif dalam UKM Paduan Suara Mahasiswa ITS, dan menjadi sekretaris 1 di UKM PSMITS pada tahun kepengurusan 2015-2016. Selain itu, penulis juga aktif dalam kegiatan PSMITS mulai dari konser dan lomba dalam negeri maupun luar negeri. Penulis juga berpartisipasi dalam kegiatan STATION (*Statistics Competition*) dan menjabat sebagai ketua region kota Madiun pada tahun 2015. Semasa kuliah penulis pernah melakukan kerja praktek di PT Kelola Mina Laut, Gresik-Jawa Timur di bidang *Quality Control* dan Balitjestro (Balai Penelitian Tanaman Jeruk dan Buah Subtropika) sebagai *Data Analyst*.

Dengan terselesaikannya Tugas Akhir ini, semoga dapat memberikan manfaat bagi berbagai pihak. Segala saran dan kritik yang membangun selalu penulis harapkan untuk kebaikan ke depannya. Penulis dapat dihubungi di akun media sosial seperti berikut.

Facebook : Fransiska Kristin Damayanti
Line ID : kristindam
Instagram ID : fransiskakristin
e-mail : kristindamay@gmail.com